

Interactive Digital Fabrication Machine Control Directly Within a CAD Environment

Frikk H Fossdal
ffo@hvl.no
Western Norway University of
Applied Sciences
Bergen, Norway

Rogardt Heldal
rohe@hvl.no
Western Norway University of
Applied Sciences
Bergen, Norway

Nadya Peek
nadya@uw.edu
Machine Agency
University of Washington
Seattle, WA, USA

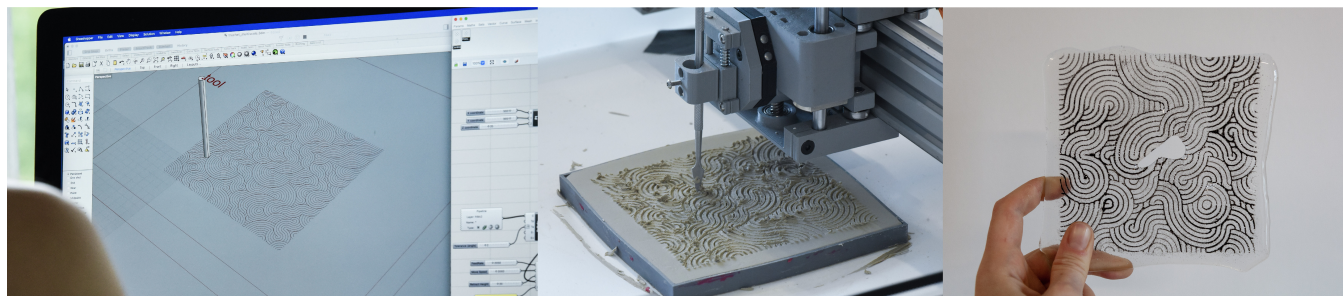


Figure 1: Our system enables direct control of digital fabrication machines from a digital design/CAD environment. Left: toolpaths are visualized in CAD and controlled from the CAD scripting environment. Center and right: Evaluation of the system by a professional glass artist. Shown here is the carving of a mold for glass tiles.

ABSTRACT

Interactive fabrication aims to close the gap between design and fabrication, allowing for rich interactions with materials and reflection in action. Drawing from craft practice, we contribute software that enables real-time control of digital fabrication machines from a Computer-Aided Design (CAD) environment. Our software not only allows interactive control of toolpath geometry, but also enables the control of machine parameters such as speed, acceleration, or jerk. This creates new opportunities for toolpath and material exploration. We evaluate our software with a professional glass artist on a custom digital fabrication machine that can accommodate multiple tools such as brushes, engraving bits, or microscopes. Finally, we reflect on implications for machine control.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI); Collaborative content creation.**

KEYWORDS

Interactive Fabrication, Computational Fabrication, Machine Control, Digital Craft, CNC, CAD/CAM, Machine Design

ACM Reference Format:

Frikk H Fossdal, Rogardt Heldal, and Nadya Peek. 2021. Interactive Digital Fabrication Machine Control Directly Within a CAD Environment. In *Symposium on Computational Fabrication (SCF '21)*, October 28–29, 2021, Virtual Event, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3485114.3485120>

1 INTRODUCTION

The late 19th century Arts and Crafts movement began as opposition to the dehumanization and alienation of growing modern industry. As part of the movement, William Morris argued that machines should be abandoned where ever they interfered with the creative process, rather favoring individual artisan touch and craftsmanship [Adamson 2021, p. 120]. In the 1960s, the Whole Earth Catalog (and its slogan "Access to Tools") took a more exuberant stance towards machines, promoting computers and technology as instruments for individual production and DIY craft [Turner 2010]. This technophilia carried through to the modern Maker movement and its obsession with "Cool Tools" (including laser cutters, 3D printers, and CNC milling machines) as a way to increase access to production [Morozov 2014].

Yet the relationship between craftsmanship and machines remains under constant negotiation. In *Digital Entanglements: Craft, Computation and Collaboration in Fine Art Furniture Production*, Cheattle and Jackson detail how the fine art furniture maker Wendell Castle incorporates robots into his fabrication practice. It is not a straightforward employment, but rather a "creative repurposing, and in some way reimagining, of the tool from its expected or proprietary roots." [Cheattle and Jackson 2015]. The milling robots and 3D scanners in Castle's studio are not "turning data into things and things into data" in the straightforward way Gershenfeld describes

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SCF '21, October 28–29, 2021, Virtual Event, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9090-3/21/10.
<https://doi.org/10.1145/3485114.3485120>

[Gershenfeld 2015]. Rather, they are part of a complex work process that involves wood, glue, studio members, patterns, and clamps [Cheatle and Jackson 2015].

HCI research has sought to better understand the relationship between people, materials, and machines in digital fabrication. Especially as costs of digital fabrication machines lower, the profile of people with access to these tools changes. A broadening user base, including amateurs, artists, designers, or small companies, has a more diverse set of requirements and needs than their industrial counterparts. Novices and newcomers to fabrication represent a growing group of makers working in different environments such as shared makerspaces [Hudson et al. 2016]. This creates opportunities for HCI researchers to explore “*maker software to go with maker hardware*” [Schmidt and Ratto 2013], such as using existing objects as a design starting point [Carrington et al. 2015; Follmer et al. 2010], reusing components of digital designs [Hofmann et al. 2018], facilitating more complex design and fabrication workflows [Fuge et al. 2015; Larsson et al. 2020; Roumen et al. 2018], or scaffolding makers in combining different tools when realizing a project [Leen et al. 2019; Schoop et al. 2016].

However, not all newcomers to digital fabrication are novice fabricators. Expert craftspeople’s adoption and use of digital tools is completely different than that of novices, and also of interest to HCI researchers. For example, Jacobs and Zoran collaborated with Ju’hoan craftspeople in the Kalahari desert merging digital design and fabrication tools with a traditional ostrich eggshell craft [Jacobs and Zoran 2015]. In *Craftspeople as Technical Collaborators: Lessons Learned through an Experimental Weaving Residency* [Devendorf et al. 2020], Devendorf et al. detail how they created a collaboration between HCI researchers and a textile craft expert to be able to more deeply explore the material knowledge needed for developing new smart textiles. HCI researchers and expert craftspeople have explored many other domains, including developing ways of exploring smart home data through ceramics [Desjardins and Tihanyi 2019], core memory through quilting [Rosner et al. 2018], and electronics through carving, sewing, and painting [Buechley and Perner-Wilson 2012]. This research shows the broadness in types of expertise people arrive at digital fabrication with: where expertise in fabrication doesn’t imply familiarity with automation just as expertise in CAD doesn’t imply familiarity with robots. We therefore argue that developing digital fabrication tools requires a nuanced understanding of the practitioners who will use them. We are specifically interested in how the workflows of fabrication and craft experts might influence the development of digital design and fabrication tools, including software and machines.

The canonical digital fabrication workflow can be cumbersome, requiring makers to traverse several different pieces of software and hardware. For example, one of the most common digital fabrication workflows, FDM 3D printing, requires 3D modeling (CAD), an exchange format export (e.g. .stl), slicing (CAM), G-Code export appropriate for the machine used, and machine execution (CNC). Other digital fabrication equipment requires similar steps. As a result, the feedback loop from developing functional machine code to evaluating a fabricated result is lengthy. This slow feedback is in contrast to the direct manipulation of output preferred in many craft workflows [Jacobs et al. 2017]. This friction is demonstrated by major differences in craft workflows employing robots and how

computer controlled equipment is ‘typically’ used [Cheatle and Jackson 2015; Twigg-Smith et al. 2021]. Yildirim et al. furthermore showed that this friction is not limited to craft practitioners, but is felt by manufacturing professionals using digital fabrication tools more broadly [Yildirim et al. 2020].

When introducing the concept *interactive fabrication* [Willis et al. 2010], Willis et al. sought to understand how real-time input to fabrication machines could make new types of making and craft possible. We seek to extend this space to understand how interactive digital fabrication workflows might be used by expert craftspeople with established fabrication practices. By moving the making itself closer to the digital space where the design intent is captured, we can create new types of interactive fabrication. Expert craftspeople have nuanced understandings of materials, the marks of tools, and how actions like brushstrokes influence the work. However, experimenting with attributes like speed or acceleration of a brushstroke on a computer controlled tool is not at all straightforward. Therefore, we seek to create a system where the state, scope, and limitations of the digital fabrication machine itself is clearly communicated and can be directly used when iterating on work.

To explore this, we developed software tools for interactive control of digital fabrication machines that enable craftspeople to directly manipulate toolpaths themselves from a CAD environment. We contribute:

- Software for visualizing current machine state in real-time in CAD
- Software for configuring, creating, interacting with, and executing programs (i.e., fabrication operations such as carving or drawing) on a machine
- An evaluation of our software on a custom digital fabrication machine with Patricia Sichmanova, a professional glass artist.

Our paper is organized as follows: Section 2 provides an overview of prior work exploring craft and machine design and control in HCI and how our contributions are in conversation with this bigger picture. Section 3 gives detail of the software tools we developed for controlling machines, including parameters such as acceleration or jerk, directly from Grasshopper with visualizations in Rhino. Section 4 provides detail on the implementation of our custom digital fabrication machine with interchangeable tools that can be controlled with these software tools. Outcomes of our evaluation through a workshop with a professional glass artists is described in Section 5. We discuss the benefits and drawbacks of our system in Section 6, providing recommendations for future work creating interactive digital fabrication machines.

2 RELATED WORK

2.1 Craft and HCI

Using the tools we contribute requires craftspeople to have existing expertise in digital design. We argue that expertise with digital design tools and CAD is widespread in craft communities, especially among professional practitioners. Some HCI research has successfully employed craft paradigms as a way to engage novices in computing, e.g., by combining papercraft and circuits [Qi et al. 2015] or clay and CAD [Dave et al. 2013]. However, this can lead to associations of craft as traditional, or low-tech, and computing

as novel, or high-tech. Devendorf et al. critique this dichotomy in their experimental weaving residency, where they create a flat team of craftspeople and engineers to explore the possibilities of smart textiles [Devendorf et al. 2020]. In *What We Can Learn From Visual Artists About Software Development*, Li et al. demonstrate that artists have deep technical insight into their processes and regularly develop and iterate on software tools [Li et al. 2021]. This prior work demonstrates how craft knowledge can be deep, nuanced, and technical. Furthermore, craft-based inquiry can enable researchers to explore areas and solutions that would not have surfaced with approaches centered around problem solving [Frankjær and Dalsgaard 2018].

Computational fabrication processes can also be part of longer craft workflows that employ various other digital or non-digital steps. These types of craft workflows are supported in various HCI systems contributions. For example, EscapeLoom leverages digital fabrication to make weaving tools for new kinds of weaving [Deshpande et al. 2021]. ProxyPrint explores using digital fabrication to create fixturing objects for wire-wrapping craft [Torres et al. 2016]. JigFab is a system that uses digital fabrication including 3D printing and lasercutting to create jigs and forms that can facilitate making with non-digital woodworking power tools [Leen et al. 2019]. Takahashi and Kim explore how combining 3D printing and 3D doodling pens can allow for makers to negotiate exploration and efficiency in creative practice [Takahashi and Kim 2019]. Goveia da Rocha et al. explored ways to facilitating using multiple digital fabrication processes in sequence, especially with multiple participants [Goveia da Rocha et al. 2021]. We similarly see our contributions as part of a larger craft workflow which may contain digital and non-digital steps.

With the deep material and process expertise associated with experience and craft, alongside computational insight, researchers have also developed completely new possibilities for making. For example, Narayanan et al. created a system that can translate any 3D mesh into automated knitting instructions, making the production of complex knit surfaces possible [Narayanan et al. 2018]. Magrisso et al. developed a system for creating 3D printed joints that allow for woodworking with non-standard angles and forms [Magrisso et al. 2018]. Researchers have developed ways to program the deformations that are caused by internal stresses created during 3D printing, enabling the design of shape-shifting structures, also known as 4D printing [Gu et al. 2020; Wang et al. 2018]. Other systems create ways of making textured carving tool paths [Ma et al. 2020], incorporating non-uniform materials like tree branches in precise computer-controlled fabrication processes [Larsson et al. 2019], or aligning structural elements with the object's stress field [Arora et al. 2019]. These novel workflows arise from deep interdisciplinary insight ranging from computational to material to process-driven. With our contributions we hope to make exploring these types of workflows easier.

2.2 Machines for Novel Digital Fabrication Workflows

HCI researchers have explored modifying machines or creating custom machines to enable novel digital fabrication workflows. For

example, they have modified existing equipment to combine electrospinning and 3D printing in a single machine [Rivera and Hudson 2019], print with live cells [Yao et al. 2015], add circuit printing and pick-and-place functionality to a laser cutter [Nisser et al. 2021], or creating a workflow for stacking cut fabric to 3D print soft objects [Peng et al. 2015]. They have also created novel machines to explore new kinds of workflows. xPrint [Wang et al. 2016] and Popfab [Peek and Moyer 2017] explore the possibilities of multi-headed machines that incorporate multiple processes. FreeD [Zoran and Paradiso 2013], position correcting tools [Rivers et al. 2012], and Turn-by-wire [Tian et al. 2019] negotiate precision control and real-time interaction in the fabrication process. Some of these machines explicitly are designed to incorporate craft processes. For example, Matchsticks is a CNC machine specifically designed for the craft of wood joinery [Tian et al. 2018]. This prior work demonstrates broad interest within the HCI community to develop novel machines and workflows, including for craft processes. We strongly concur that existing off-the-shelf and industrial equipment doesn't necessarily serve the diverse group of people who might benefit from using computer-controlled processes. Therefore, it is valuable to reconfigure and redesign digital fabrication tools with different use cases in mind.

Kim et al. outline several speculative future scenarios of people engaging with digital fabrication machines under the umbrella *Human FabMachine Interaction* [Kim et al. 2017]. They use these speculative futures to critique the current unidirectionality of digital fabrication machines. They argue that future digital fabrication machines should support mixed craft practice, should allow for design decisions to be made during the fabrication process, and should enable simultaneous and real-time collaboration between the human and machine. Concretely, they argue that Human FabMachine Interaction should enable adaptive and real-time processes while supporting iteration. We seek to explore these goals by creating a new and experimental controller that allows for the iterative and creative exploration of precise parameters offered by the machine.

Developing machines and machine controllers requires expertise from different fields. Prior research has sought to lower the barrier to machine building itself to enable a proliferation of diverse applications. The Cardboard Machine Kit provides individual modules that can be combined in different ways to make up the motion system that can be outfit with a custom end effector [Peek et al. 2017]. Fabricatable Machines are parametric designs of machine modules that can be customized to create different machines [Fossdal et al. 2020]. Jubilee is a 'mostly-printed' machine with automated tool-changing that users can extend with custom tools [Vasquez et al. 2020]. Clank is a suite of design patterns for 'mostly-printed' open-source hardware machines [Read 2021]. We draw from these machine toolkits, especially Clank, when developing our own machine testbed for our interactive control software.

Beyond machines themselves, there has been limited research into creating frameworks for interactive machine control. The three examples shown in the original *Interactive Fabrication* paper use a touchscreen, voice, and gesture as direct input [Willis et al. 2010]. This allows for direct manipulation using these input modalities. Constructable [Mueller et al. 2012] allows laser cutter users to input instructions directly to a machine using a laser pointer as input.

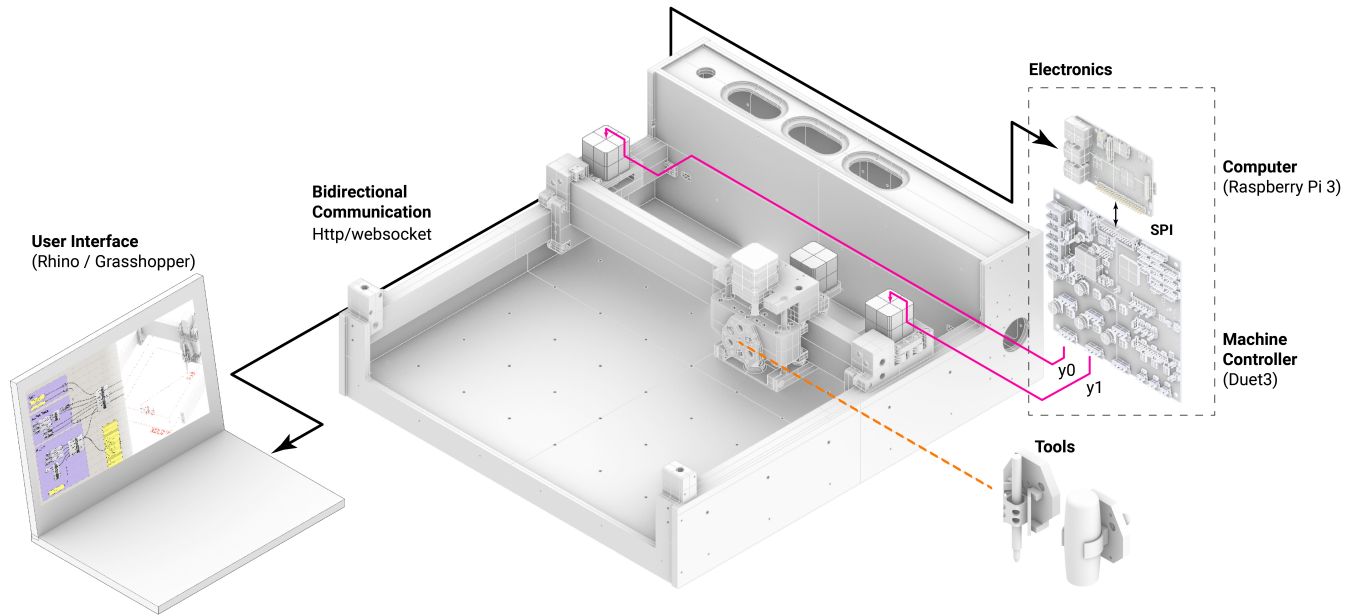


Figure 2: System Overview: Our system enables an interactive workflow between a digital fabrication machine and the CAD environment. The core of our implementation is software components implemented in the popular programming language Grasshopper that enable real-time communication between Grasshopper and a machine controller via http requests and websockets (left). We created a machine with interchangeable tools to test different fabrication workflows using our system (right). Our implementation employs an off-the-shelf Raspberry Pi and Duet3D controller and can be readily adapted to other machines using the same electronics.

However, rather than create new input devices, which require experience to gain proficiency, we seek to augment the existing CAD environments that craftspeople are already using with direct machine control. A closely related example is LINC, or Live Interactive Numeric Control [Li et al. 2017]. LINC is a sketch-based digital design tool that enables artists to directly output their tablet-sized digital brushstrokes to a large-format CNC machine. Evaluating LINC, Li et al. found that linking the artist and machine more directly enabled rich interactions absent from traditional digital fabrication including exploration, reflection-in-action, and the ability to respond immediately to material outcomes. We believe that these rich interactions are partially made possible due to the LINC drawing environment being quite similar to other vector drawing programs such as Inkscape or Adobe Illustrator that the artists regularly use in their practice. We hope with our contributions to make similar rich interactions possible, but rather than starting from a digital drawing program, starting from CAD. Furthermore, we extend the possibilities that LINC explored by contributing controllers that not only interpret geometry, but also take in parameters for the machine’s choreography, including toolpath speeds, acceleration profiles, and jerk.

3 INTERACTIVE MACHINE CONTROL FROM CAD

The core of our system is a collection of tools that enables users to communicate and interact with a machine directly from a CAD program. As described in the introduction, the canonical workflow

that digital fabrication tools typically enforce requires users to traverse several data formats and softwares in order to transform their machine code to fabricated results. We want to shorten these steps by providing a collection of software tools that allow users to operate within one single space where both the digital design and the physical making are much more tightly integrated.

Although digital fabrication machines offer different workflows, their operation is often similar. They consist of mechanical assemblies that use electromotors to move and position a tool (often referred to as an end-effector) within the boundary of the machines operating space (the machine envelope of a machine). These motors are connected to driving circuitry that translates Cartesian coordinates (x,y,z) into electrical impulses that move the motors. In most modern machines this circuitry is integrated into a machine controller. In order for our users to be able to both reconfigure properties of machines and directly execute programs from a CAD program, we have to implement communication interfaces directly between the machine controller and the CAD program.

Grasshopper is a programming language for creating parametrically driven designs, and is offered as an extension to the popular CAD tool Rhino3D. Grasshopper includes a number of off-the-shelf tools to manipulate geometry data through user-defined dataflow graphs. As users create Grasshopper programs, the output is constantly visualized in the CAD workspace of Rhino. By enabling tools in Grasshopper to communicate in real-time with a controller, we can both visualize the properties of the controller in Rhino and send programs made in Grasshopper back to the controller.

From this context we derive the following design goals:

Creative Control: We want to enable our users to easily manage and create *any* raw toolpath and be able to run these directly on the machine from their scripting environment. To that end our implementation needs to incorporate both logic for converting drawn geometry in Rhino/Grasshopper into functional machine actions as well as logic that enables users to execute machine actions directly from Grasshopper.

Interaction: Above all, the implementation needs to communicate how user intention is perceived by the machine controller as well as the limitations of the machine to the user. To enable this, we need ways of communicating the state of the machine in real-time to the user and visualization tools for how the machine interprets a user's machine code. However, we do not want the implementation to limit the user's ability to provide input. The components are implemented to provide the minimum connection that allows a user to interact with all controllable parameters of the machine. It is up to the user to create the type of interface and in canvas interaction that they want. Interfaces for executing programs and jogging the machine can be simply added by adding off-the-shelf components from Grasshopper like sliders, buttons, or toggle switches.

Turn taking: We see this workflow as a continuous dialog between the machine and the user. When a user executes a program from Grasshopper, the machine will run the program and report back its progress in real-time. However, this should not limit the user in continuing working in Rhino / Grasshopper. To that extent, our implementation should enable an asynchronous interaction, where the communication interfaces towards the machine do not halt or block the user interface in Grasshopper. The user should also have the ability to pause and stop a running program from the Grasshopper environment. Finally, the created workflow should be bidirectional. As data is collected by the machine, the data should be piped back into the Rhino canvas and made available for the user.

Based on these initial design goals we have implemented two components in Grasshopper that facilitates an interactive workflow between the CAD space and a machine controller:

- *The Controller:* a component that act as the physical controller's virtual counterpart. Through this component, users can run and execute machine programs, reconfigure the properties of the controller, and access and visualize important properties of the controller like position of the end-effector and the work envelope of the machine.
- *The Processor:* a component that converts geometry in Grasshopper into executable programs that can run on The Controller.

We have chosen Grasshopper due to its popularity within art, design, and engineering. Both of the components are written in C# using the provided Visual Studio templates provided by McNeel [McNeel and Associates 2021].

The system is designed to operate in unison with the popular machine controller Duet 3, coupled with a Raspberry Pi [Duet3D 2021]. We have chosen the Duet ecosystem due to its relative flexibility in type of machine configuration. It allows users to remap its inherent configuration, allowing it to work with different types of machine configurations. The Duet3 board has 6 driving circuits for

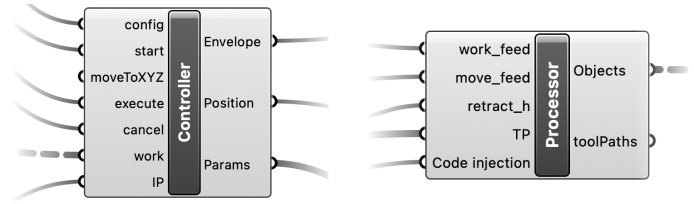


Figure 3: We have implemented two custom “components” in the Grasshopper programming environment, *The Controller* and *The Processor*. Combined, these two components allow users to have interactive control of a machine directly from their CAD environment. Users can combine these components with Grasshopper’s existing components for defining and manipulating geometry to create custom machine control “definitions” aka Grasshopper programs. The description for the different input and output parameters are in Table 1 and Table 2.

stepper motors, and allows users to remap the kinematic arrangement of these motors and their connection logic to different sensors. Additionally, the controller has several optional IO ports that can be used to configure different sensors and external embedded logic, such as limit switches, temperature sensors, etc.

In the following subsection we detail the functionality and implementation of these components.

3.1 The Controller

The Controller is a virtual representation of a physical machine controller. Through this component, users can input movement commands and fabrication programs that can be executed on the real controller. Simultaneously, the component monitors the state of the controller, feeding back parameters such as current position of the end-effector or the status of the machine (if it is busy or idle). Additionally, The Controller lets users send configuration commands that change the inherent configuration of the controller. The implementation is influenced by other similar real-time implementations in Grasshopper [del Castillo 2018; Payne and Johnson 2010; Tahanzadeh 2015], but differs in its specificity towards digital fabrication platforms and in its use of parallel threading.

To enable this type of interface in Grasshopper we had to go through several different steps in our implementation. These are detailed in the following subsections.

3.1.1 Communication. The Duet3 controller offers several communication protocols for interacting with it. Our implementation relies on the DuetSoftwareFramework, which enables the Duet3 to be controlled from an attached Linux based mini computer (in our case the Raspberry Pi 3). The DuetSoftwareFramework includes services for communicating with the Duet3, including a WebSocket connection point to monitor the state of the controller and HTTP endpoints that allow us to send commands to the controller. Our system communicates with the controller over TCP/IP; the user specifies the network location of the controller by using the IP input of the Grasshopper component.

3.1.2 Monitoring and Visualizing state. The Controller incorporates a local class structure that stores the real-time state of the

Parameter	Type	Description
config	input	Configuration parameters of the controller
start	input	Start communication interface with the machine controller
moveToXYZ	input	Move to a specific position
execute	input	Execute work on controller
cancel	input	Cancel all currently running work
work	input	Machine code to be executed
IP	input	IP adress of the controller
Envelope	output	The work envelope of the machine represented as a Rhino Brep Box.
Position	output	The tool position perceived by the controller.
Params	output	The state model of the controller. Represented as a list of text objects

Table 1: The input and output parameters of The Controller

Duet controller as it is received. The Controller connects to the controller via WebSocket and receives updates from the Duet board each time the state changes. This includes all the properties of the Duet like the position of the individual axes, if the axes are homed, current status of the machine (if its busy or idle, current defined work envelope and current position of the end-effector). The Controller makes this data available as a list of text objects on the Params output. This allows users to extract the specific data they are interested in and create sub-routines that use the output to visualize properties in real-time in Rhino. This enables users to, for example, import a CAD model of their specific machine and use the outputted data to map the different axes of the virtual CAD model to the corresponding real-time parameters of the respective axes. Thus users can incorporate their own machines and visualize them in real-time. We have also chosen to expose two specific parameters from the state model: the end-effector position and the work envelope of the machine. This is to enable users to quickly visualize two of the most important features of the controller.

3.1.3 Sending Commands. Commands are sent to the controller via HTTP post requests. On initialization the component initializes an empty local buffer of movement commands, or Work Objects (Work Objects are explained in the next subsection) As Work Objects are inputted to The Controller the buffer is populated with Objects to be executed. The component includes logic to ensure that the machine is in a safe state before movement commands can be parsed and executed, meaning that the machine needs to be homed before The Controller will execute any Work Objects. Additionally the component checks the status of the machine controller (busy or idle) before sending a move object. If the status is busy, it will not send command and wait for a set interval. Once a Work Object is sent, the status of the controller will be set to busy, and new movement objects will not be sent before the status of the machine is set to “idle” again. This logic ensures that a user can send a cancel command between the execution of each Work Object.

3.1.4 Configuring Control. The Controller contains an input that lets user input configuration commands to the controller. The Duet uses M-Codes to configure its properties and it stores its default

configuration in a local file on the Raspberry Pi. This configuration file is read on each start-up of the system. The Controller currently does not hold logic to redefine the default configuration of the Controller, rather, it lets users inject new configuration commands on top of the old configuration. If the user wants to overwrite the default configuration, they can extract the configuration parameters they have set in Grasshopper and add them to the file either through the web-interface that the Duet provides or by manipulating the file directly. For example, a user can change the speed and acceleration parameters of the individual motors that are attached to the controller, or change how the the different motors form a kinematic chain (how they are positioned in relation to each other).

3.1.5 Multi-threading in Grasshopper. Having communication interfaces that await replies from external sources poses a problem in Grasshopper. Grasshopper is implemented as a GUI running on a single thread on the computer. Calling state via WebSockets and sending commands over HTTP requests will inherently block the Grasshopper GUI as it is waiting for replies. To avoid this problem The Controller is running the communication interfaces on two parallel threads that run in the background; it only invokes the GUI thread when necessary. One thread monitors the WebSocket connection and updates the local state model of the controller on each received update. This will trigger the component to update its output parameters, making the latest received data from the controller available for the user through the component output: Envelope, Position, and Params. The other thread monitors the queued Work Objects in the local buffer. When the user triggers the execute input of the component, the thread will check if the controllers status is idle and start to execute the Work Objects and remove them from the buffer as they are executed.

3.2 The Processor

The Processor takes user generated curves and converts them into encapsulated blocks of executable machine code, or what we call

Parameter	Type	Description
work_feed	input	The feedrate of the machine when working on a toolpath
move_feed	input	The speed of the machine when moving from a toolpath to the next
retract_h	input	How much the tool should retract before moving from a toolpath to the next
TP	input	The toolpaths to convert. Can be inputted as a single object or a list of objects
Code Injection	input	Custom gcode to inject at the top of a machine program
Objects	output	Work Objects to be executed
toolPaths	output	All moves represented as a list of lines that is used for visualization purposes

Table 2: The input and output parameters of The Processor

Work Objects. A Work Object represents an operation that the machine will perform, and is in the G-code format. Typically a fabrication program will consist of many Work Objects. For example, if the user wants to execute a series of polylines, the generated chain of Work Objects will be 1. retract the tool i in Z direction, 2. move to the first point of the first polyline, 3. move tool down in Z direction, 4. execute the polyline, 5 retract tool, 6. move to next polyline, and so on.

The Processor component takes toolPaths, feedrate, moveFeedrate, retractHeight, codeInjection and either single curves or sets of curves as arguments, and loops through these curves and converts them into raw G-code. The component outputs the G-code as lists of strings structured in a Grasshopper Data Tree structure. Additionally, the component outputs a list of curves that can be used to visualize the toolpaths in the Rhino Viewport. The Processor also holds logic for sorting all the Work Objects so that they are executed in an left-to-right order.

The CodeInjection input allows users to input custom G-code commands that will be injected between each curve operation and added to the data tree as raw G-code. This allows the user to add

commands specific to their machine implementation, e.g., controlling the speed of a cooling fan. The CodeInjection takes either a single text value, or a list of text values. These are typically generated using the panel block that is an off-the-shelf component in Grasshopper.

4 FABRICATABLE TABLETOP MACHINE WITH INTERCHANGEABLE TOOLS

We implemented a small desktop-sized machine to evaluate our system in different workflow scenarios. We needed a machine that could facilitate on-the-fly experimentation where we had full control over the machine controller, the mechanical assembly, and the different tools that were mounted on to it. The machine was specifically designed to accommodate the use of different tools. Its main feature is a magnetic tool-changer that lets users manually swap tools. Another important motivation behind the implementation was to make the machine small enough to fit on a desktop so that it could be easily accessible and used on a day-to-day basis.

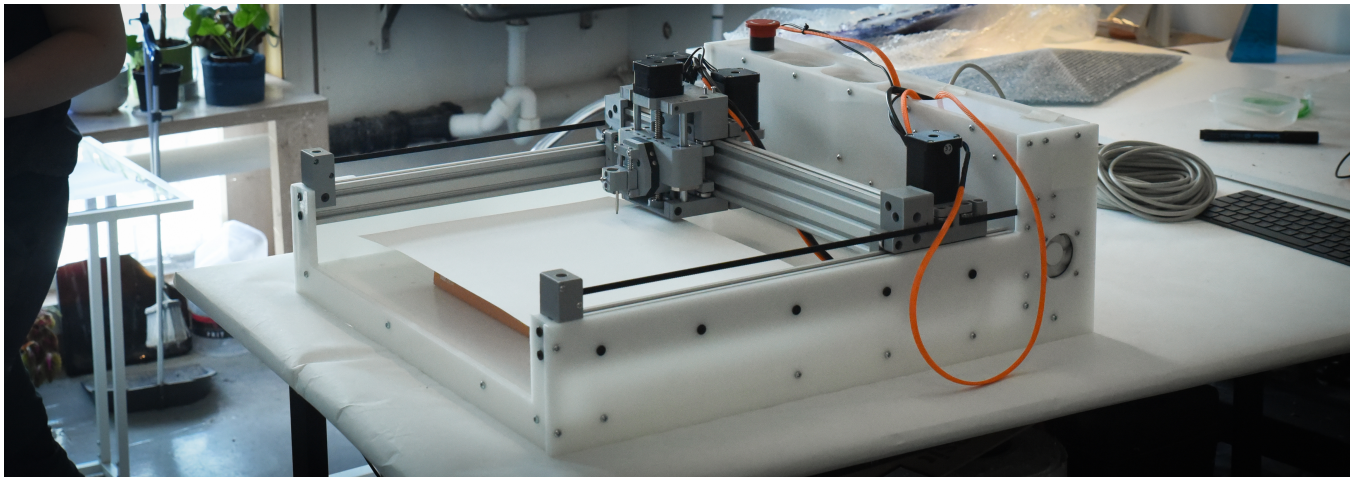


Figure 4: We made a small desktop sized machine with a magnetic tool-changer that could facilitate a broad range of experiments using our system. The machine is partially based on design patterns from Clank [Read 2021]. This machine provided a portable testbed for evaluating our software which we employed in our own tests and in a workshop with a glass artist.

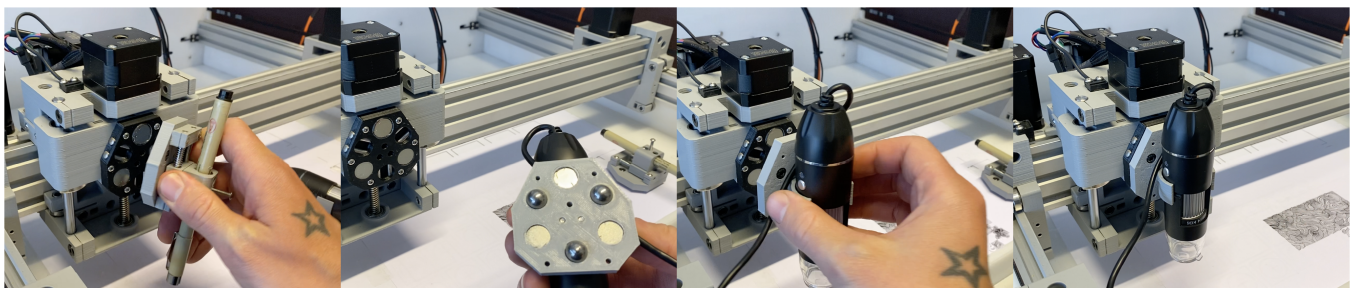


Figure 5: A magnetic toolchanger on our machine to allows us to easily swap tools and work with different workflows. Shown here is swapping from a pen-holder tool to a microscope tool. We implemented a number of other tools including a spring-loaded holder for carving tools and a small spindle for milling.

In this section, we provide detail about the design of the machine and how we used our Controller and Processor components to enable interactive workflows on the machine.

4.1 Design

Figure 2 shows an overview of the machine design together with its overall system integration. Here, a Duet3 controller is coupled with a Raspberry Pi to control the different actuators and sensors of the machine. The core machine design is based on the the open-source hardware Clank design [Read 2021]. The majority of the parts used in the Clank design are parts that can either be printed with a FDM 3D printer, or parts that can be easily sourced such as stepper motors, fasteners, and aluminum extrusions.

Our machine is shown in Figure 4. The machine is designed to accommodate a work envelope of 350x350x50mm. To make the machine more mobile for field experimentation, we encapsulated the machine in a CNC milled POM frame that encloses the power supply, Raspberry Pi and Duet3 controller in the back of the frame. Additionally the machine was fitted with an emergency button, and a soft-stop button that resets the system and cancels all movement.

An important attribute of the machine is its ability to accommodate different tools, or end-effectors, enabling users to work with the different workflows that these tools entail. To enable us to easily swap tools and experiment with different end-effectors, we modified the original Clank design with a custom designed Z/X-axis carriage that incorporates a custom toolholder. The design of the toolholder uses a kinematic coupling design pattern to allow different tool-heads to be attached to it. The use of this type of design pattern of a kinematic couplings is motivated by similar implementations such as the Jubilee design [Vasquez et al. 2020]. An

example of tool swapping can be seen in Figure 5. Here we can see a user swapping from a pen or plotting tool to a USB-microscope.

The magnets allow us to manually swap tools without using any fasteners. The kinematic coupling of the toolchanger ensures that the position of the tools will be the same for any tool. This is important so that users do not need to re-calibrate the machine as they swap tools. Similarly, we made a general toolPlate design that can be incorporated into designs for different tool-holders, e.g., for holding brushes, carving tools, etc. As the majority of the applications and workflows the machine will be used for are not torque intensive, we are able to keep the tools in place using three strong disc magnets (approx force 44N) to preload the tool. For more force- or vibration-intensive applications (such as milling) we added threaded holes on the toolholder that allow users to use fasteners to secure the tool.

We have made several tools for the machine. These include a spring-loaded toolholder (for pens, markers, knives), a microscope tool, and a touch probe that can be used to probe surfaces. The tool-holder is designed with the goal of being easily fabricated using a FDM 3D printer and easily sourced parts.

4.2 Creating an interface for the machine

We used our system to implement a real-time control interface to our machine in Grasshopper. Figure 6 shows the interface in Grasshopper (left) and its output in Rhino (right). We structured the different interfaces in appropriate groups (purple) to make the interface more intuitive.

The **Config** group allows users to input configuration commands in the form of M-codes that are read by the Controller, as explained

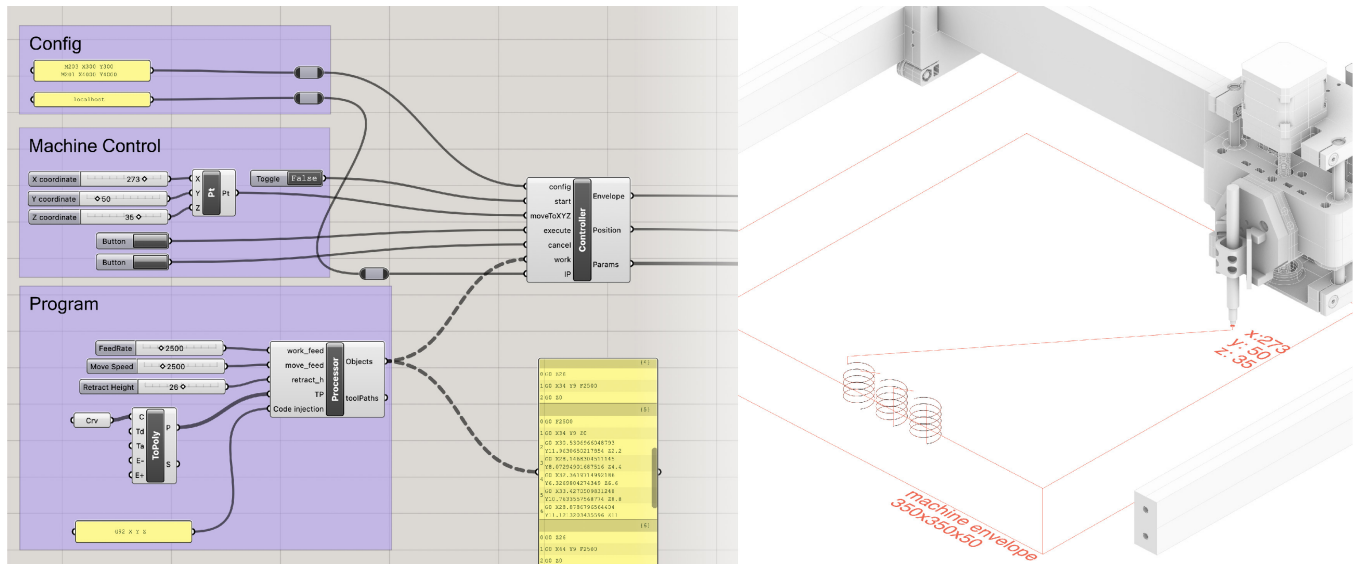


Figure 6: The machine interface. It uses The Controller component to communicate and interface with the machine (Grasshopper canvas, right), and The Processor component to generate Work Objects that can be executed (Grasshopper canvas Program group, left). The visualized output of the system can be seen on the right. Here we can see how the position of the tool is being visualized in real-time together with the defined work envelope of the machine. The visualized Work Objects (here the spirals) come from The Processor.

in Section 3.1. This is also where users define the IP address the controller is located on. In our case the IP is set to *localhost*.

The **Machine Control** group lets users activate The Controller's communication interface, execute and cancel programs, and jog the machine. The jogging point is made using the off-the-shelf *Construct Point* component in Grasshopper. It outputs an encapsulated point in the Rhino format *Point3d*. The Controller will execute a movement to this point on each triggered change. When triggering the Execute button, the Controller will execute all Work Objects that are buffered in its local buffer. The Cancel button cancels the Controller's current Work Object and empties the buffer.

The **Program** group lets users create work objects that are to be executed on the machine using The Processor component. In the example in Figure 6, tree spirals are imported from the Rhino workspace and converted to polylines using the off-the-shelf *Curve to Polyline* component in Grasshopper. *Curve to Polyline* lets the user set the interpolation resolution of curved lines. The Polyline is then fed to the Processor together with *work_feedrate*, *move_feedrate* and *retract_height*. The output of the Processor can be seen in the yellow text box to the right of Figure 6. Here we see raw G-code commands structured in a Grasshopper Tree Structure. This data is piped into The Controller and is ready to be executed on the machine.

This demonstrates how we can use our software components Controller and Processor to create a custom interface for a machine in Grasshopper. Using the virtual Grasshopper Controller as a connection point to the physical Duet controller, we can define our own logic and interface to interact with the machine. Simultaneously, we can connect our own logic to visualize different properties of the state of the machine (in this case the machine envelope and the tool position). This enables us to incorporate the physical machine as a part of our Grasshopper canvas. As we are developing definitions and designs in the Grasshopper canvas, we can connect the definitions to the program group and execute them on the machine. As we are evaluating the machines performance in regards to our design, we can change the configuration parameters through the configuration interface.

5 INTERACTIVE WORKFLOWS

5.1 An Example Workflow: Plotting and Refining a Trouchet Tile Pattern

To showcase the type of interactivity that our system provides, we present a specific case where we use the implemented Grasshopper components and our implemented machine to plot a Trouchet tile pattern on a sheet of paper using a pen-tool, then examine the plot using a microscope tool. Trouchet tiles are square tiles decorated with patterns that are not rotational symmetric. When combined on a plane they can be combined to form variations of different composed patterns. We made a definition in Grasshopper that parametrically defines Trouchet tile patterns over a given *XY* surface. The pattern is outputted as a list of polylines that can be fed into The Processor.

We use the visualized machine envelope and tool position to determine where the plot is being drawn on the physical machine relative to the visualized machine envelope in Rhino. This information helped us to place the sheet of paper correctly in terms of the placement of our design in Rhino. The jogging interface is then used to position the pen over the paper sheet, moving the pen down so that it contacts the paper and the ink bleed is visible on the paper, and finally setting the zero position of the tool using a injected homing command in The Processor Component. Before executing the complete Trouchet pattern, we draw a rectangle around the pattern in Grasshopper and execute it on the machine to understand where the machine will draw the pattern. This is also useful for determining if the acceleration parameters and feed speed is appropriate in terms of the workflow and the visual aesthetics we want from our plot. These kinds of plotting tweaks are extensively explored by plotting artists [Twigg-Smith et al. 2021]. Executing different programs, either the rectangle or the complete Trouchet pattern, is done by simply changing the input connection to the work input on the Controller component. As the pattern is defined as a parametric design that takes scaling on a *XY* plane as input, we can readjust the size of the pattern to fit within the rectangle.

After plotting, we changed the tool of the machine to a usb-microscope tool. To calibrate the focus of the microscope we jog

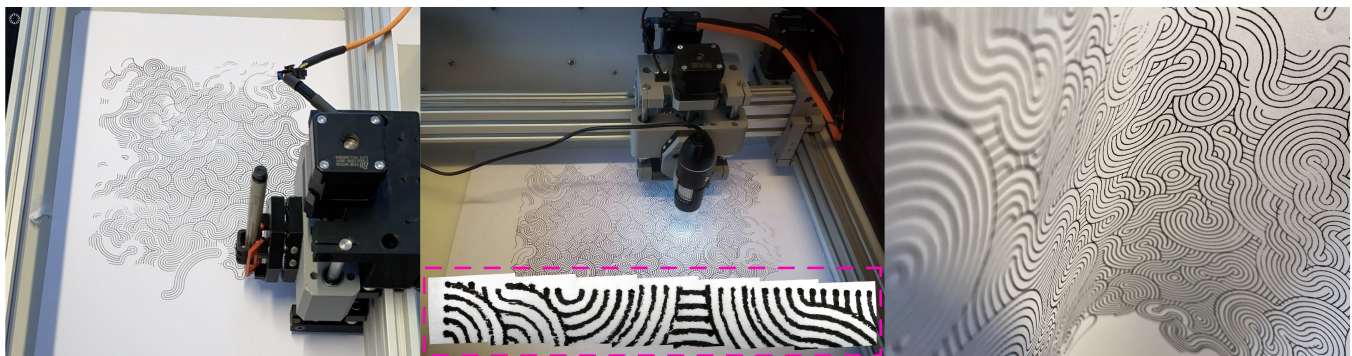


Figure 7: Using the system to plot a Trouchet tile pattern on a piece of paper and scan the pattern using a microscope tool (the microscope images are shown in the dotted purple rectangle). Based on the scanned results from the microscope we lowered the speed of the plotted lines to make the marker bleed better into the paper. The finished result is shown on the right.

the Z-axis using the position input to The Controller while monitoring the video feed from the microscope. We then generated a definition in Grasshopper that takes images with the microscope at given intervals. The images showed us that the ink was not properly bleeding into the paper. We also saw that there were slight vibrations in the plotted lines. To get a better result, we lowered the speed of the plotting program using the speed parameters in The Processor component and reran the program with better results.

This describes a typical workflow that our tool enables. The workflow provides a continuous turn-taking practice, where the machine is used actively to inform the user about the limitations, aesthetics and accessible design space he or she can operate within. As tools are switched on the machine, we define new routines in Grasshopper and connect them to The Controller and execute them. The plotted result, together with the microscope images of the lines, can be seen in Figure 7.

5.2 Evaluating the Interactive Fabrication Possibilities with a Professional Glass Artist

To further evaluate different use cases of our system, we conducted a two day workshop with a professional glass artist, Patricia. Patricia stood out as an ideal candidate to test our system as she used different digital fabrication tools in her daily practice and had prior knowledge in the use of both CAD and CAM tools. She is also working part time as a workshop manager in both a local fablab and in a local design school. She did not have any experience in machine building nor creating bespoke digital fabrication equipment. We used the implemented machine from the prior examples as a basis for the workshop, and prior to the workshop we made a set of custom tool-holders for the various equipment she wanted to test out. Patricia was given a walk-through by one of the authors of how the different components in Grasshopper worked and how they could be used to interact with the controller. This included a

modified version of the spring-loaded pen-holder that could hold different glass etching tools, and a mount for an electric spindle.

In the beginning of the workshop she used the Controller component to simply jog the machine and understand how this connected with the visualization in Rhino. After becoming familiar with the different components she used the Processor component to create a simple program that drew a circle on a sheet of paper using the pen-tool. She proceeded to test how the machine responded as she changed the acceleration and speed parameters of the controller when executing the circle. At this point she ran into several issues, namely by setting the acceleration or speed parameters too high and making the machine go into a stall. She used this information to further understand the limitations of the machine and applying as she moved forward. She commented on the how the workflow saved her a lot of time, being able to do every step inside of Grasshopper:

"The workflow itself spares a lot of time, with no need to move into other softwares. Also visualizing the machine state straight into Grasshopper is a huge advantage"

Next, she mounted a diamond cutter to the spring-loaded tool and inserted a piece of scrap glass into the machine bed. By using the jogging interface she positioned the tool over the midpoint of the glass plate and lowered it until it touched the surface. She commented that the diamond cutter should have just the right amount of pressure, and she wanted to do a series of tests to determine what that amount should be. She used the circle program she had created to test this. First, she set the zero point to just when the diamond cutter was touching the surface and ran a circle program. She stored the midpoint in a separate program to easily move the machine back to the same point. Not satisfied with the result of the first circle, she moved the tool back to her zero point, and lowered the tool so that the spring mechanism applied more force to the tool engaging on the glass sheet. She then re-ran the circle program with an offset to a new location.

As the Grasshopper Canvas grew with different routines and customized interfaces for interacting with the controller, she started

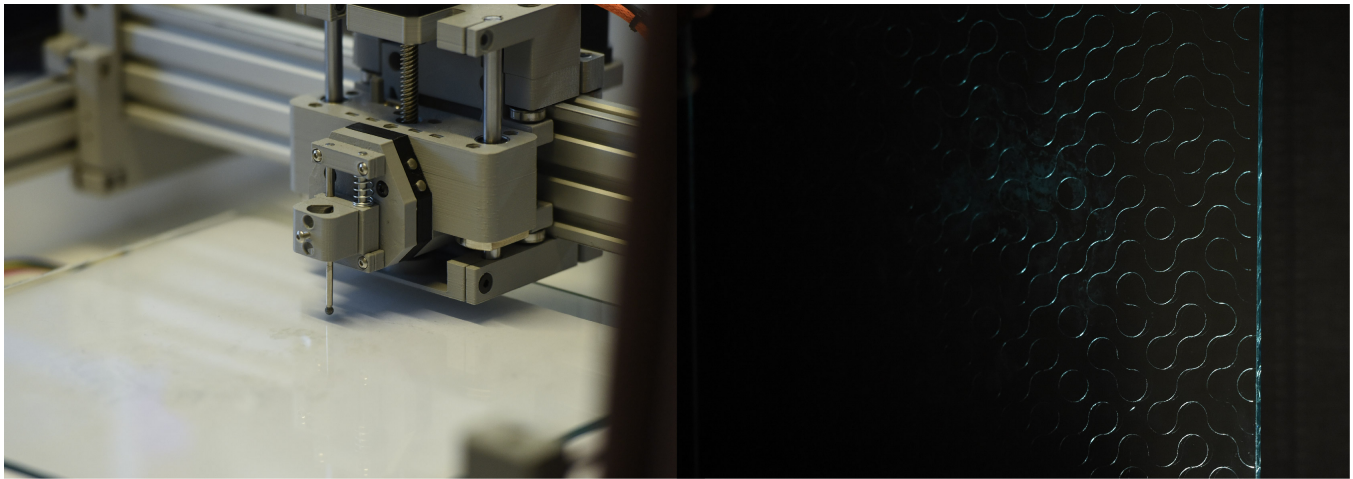


Figure 8: The artist attached a diamond cutter to the spring-loaded tool and engraved patterns in a sheet of glass. She iterated several times to optimize the amount of force exerted.

grouping the different routines and programs into Grasshopper groups and labeling them accordingly. She used this to keep track of where she had already made cuts in the glass sheet, and she commented how this structuring was a good addition to her workflow. Particularly interesting was the way she made small interfaces on the different designs. Each individual design was grouped with specific interfaces for that design. For example, one design could require specific acceleration control. Another could require more current in the motors of the machine. This would be specified on the different designs as they where needed.

Satisfied with performance of the machine, Patricia made a more complex program to engrave the glass sheet. She removed the test piece of scrap glass and inserted a clean glass sheet in the machine. Prior the workshop Patricia had made some pattern generating definitions in Grasshopper that she wanted to engrave in the glass sheet. She connected the output of these scripts to the Processor component with the already determined feeds and speeds from the circle test, and used the same Z-height for her diamond cutter tool. As the program was running and engraving the generated pattern, she noticed that the machine was not traversing the curves as she

expected. The machine accelerated in jerky and non-smooth steps through the lines. To address this issue she canceled the operation, and increased the value for the machine's jerk-parameters (*jerk-parameter*, or *jerk-speed* is the machines configuration for allowable instantaneous speed change). Rerunning the program was much smoother. The engraved pattern is shown in Figure 8.

On the second day, Patricia had prepared several new types of experiment that she wanted to test. She continued working in the same Grasshopper canvas from the previous workshop day, where all her machine program definitions and parameters for machine configuration were saved. We also prepared several additional spring-loaded tool-holders so that she could use more tools.

Patricia explored several different workflows on the machine. Based on the experience from the previous day, she engraved several mirror glass sheets with parametrically designed fonts in Grasshopper. She created a differential line growth pattern that was cut into a prepared clay mold that was to be used for molding glass. She used the Troughet tile generator from the previous section to engrave a plaster mold. As plaster and clay have very different harnesses, she had to adjust the feedrate of the program to accompany for

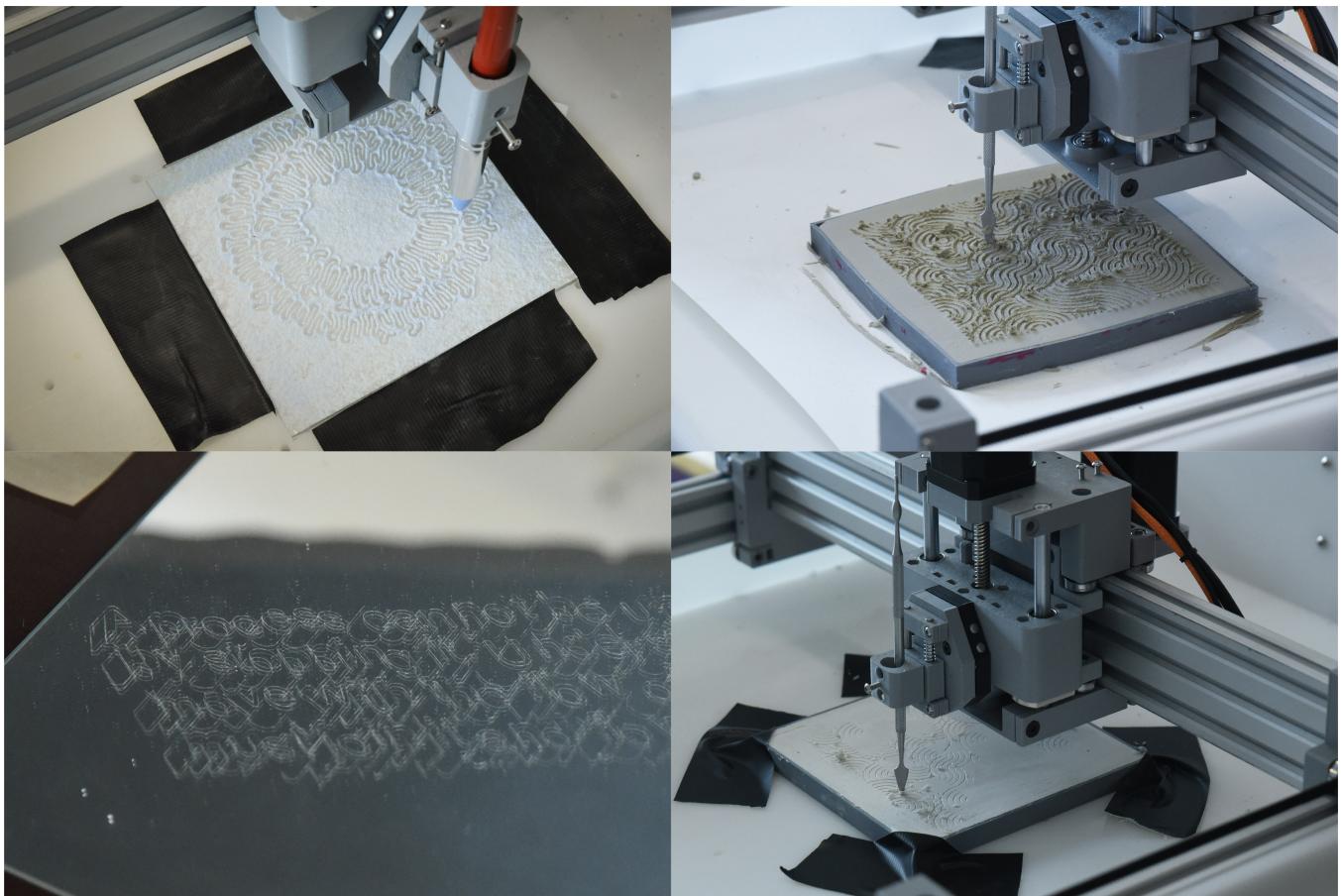


Figure 9: Four different workflows that were experimented with during the workshop. Upper left: a brush being used to imprint a pattern on a glass sheet covered with grained glass particulates. Upper right: a clay mold being engraved by a knife tool. Lower left: an isometric text engraved in a sheet of mirrored glass. Lower right: a plaster mold being engraved by a knife tool.

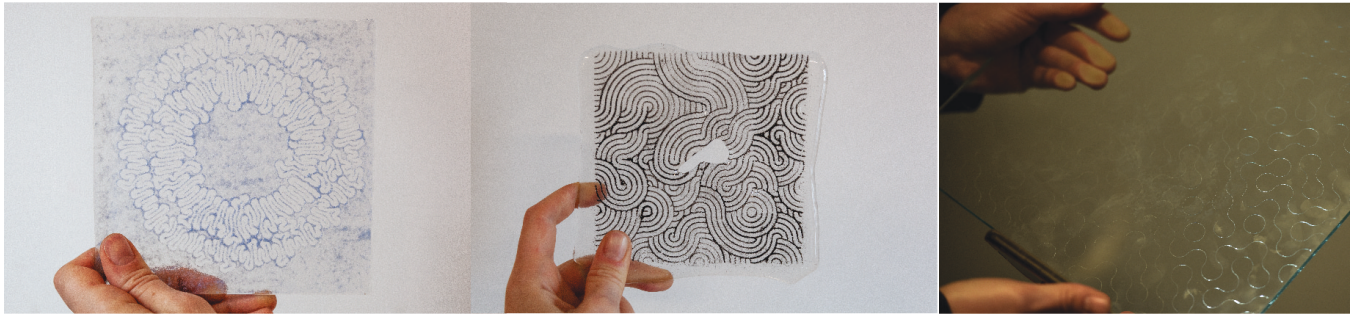


Figure 10: Results from the workshop. Left: a glass sheet covered in colored glass powder shaped with a differential line growth pattern after it has been fired. Middle: a glass plate that was molded using a clay mold engraved with truchet tile pattern. Right: a glass sheet engraved with a line pattern.

this. She commented on how the system together with the machine allowed her to experiment more freely:

"The machine is much more approachable than a huge CNC, and I can use many different workflows in my art practice. Its not as powerful as a big CNC, but the way it easily lets you change tools and fasten them on the machine makes a big difference when I'm experimenting with different aesthetics"

She also made remarks about the advantage of being able to control and draw her own toolpaths. To her, a toolpath is more than just a technique of removing or adding material. It is a visual medium. Several of the tests she did were to experiment with the visual imprint a specific toolpath can have on a object. Her latest art project was about exploring how different toolpaths affect the aesthetics of glass:

"I use toolpaths actively in my work to create visual patterns. By using glass with different colored layers, I want to increase the contrast. In the future I want to play with positive and negative space. When you take an imprint of a CNC milled piece, the opposite space is beautiful."

Overall Patricia considered the system useful for her practice. By continuously moving back and forth between the virtual design space in Grasshopper and the physical space of the machine, she was able to prototype several different workflows. She made several positive remarks about being able to design and prototype the toolpaths themselves, instead of a more traditional route where a 3D model is first created, and toolpaths are derived from the 3D model using some kind of CAM tool. To her, the aesthetic characteristics of the object she was making was embodied in the characteristics of her toolpaths. She commented that the most valuable facet of our system was the ability it gave her to prototype and test these different characteristics quickly by simply drawing it in Rhino and executing it through the components in Grasshopper.

When asked about the advantage of the workflow that our system provides, the Patricia emphasized one feature; the ability it gave her to prototype the visual aesthetics of a tool path. She coupled this with two facets of the implementation: the machine's ability to quickly swap tools and the system's ability to let her experiment with those tools in a direct way, not forcing her to traverse

a lengthy stack of different software packages. She reflected over how traditional CAM tools were black box systems that sought to mimic a virtual model in real life by removing material with toolpaths optimized to be as efficient as possible. She commented that this ultimately left the user with very little control over the visual result that these toolpaths leave on the object. To her, our system was a tool that gave her control over this aesthetic, and an ability to directly texture and tune the visual property of different objects.

By the end of the workshop Patricia had explored several different workflows and fabricated multiple different pieces. She was most excited about the different molds she had fabricated in plaster and clay. After the workshop she used the molds to mold glass tiles where the grooves imprinted on the tile was used to hold colored glass grain. This was then fired while clear glass was poured over the tile to make a solid glass base. At this stage she commented on how this was a completely new technique for her and that she had to experiment several times with different fire temperature to get the glass powder to fuse correctly with the glass base. The end results of the workshop are shown in Figure 10.

6 DISCUSSION

We sought to explore how craft practices could influence the design of digital fabrication systems. In particular, we sought to explore how expert craftspeople, many of whom are proficient in digital design, could have access to the immediacy and material control that other forms of fabrication afford. We developed software to connect the digital design world of CAD directly to computer controlled fabrication machines. We included the design goals of creative control through real-time and turn-taking paradigms. This is in line with prior research visions for Interactive Fabrication [Willis et al. 2010] and Human FabMachine Interaction [Kim et al. 2017]. We take the explicit position that introducing bidirectionality directly from CAD to the end-effector and back without intermediate translation steps enables rich forms of exploration for digital craft.

When evaluating these bidirectional machine control tools on a desktop digital fabrication machine with the professional artist, we gleaned several insights. Firstly, both her comments and workflow confirms our original instinct that different workflows in digital

fabrication is not as linear as they are often portrayed. Her process would consist of several variations and material tests before arriving at a point where she was satisfied with the result. She commented several times on how shortening this iteration cycle was enormously valuable for her practice, and how using just one software package sped up her process of experimenting with different workflows, especially different kinds of “brush strokes” which she could vary the geometry and choreography of.

Secondly, it was interesting to observe how the toolpaths themselves where used as a visual expression in her art. In most modern CAM packages, toolpaths are just a means to remove or add material in an efficient way, not necessarily a means to create the visual aesthetics of the end result. Tuning or tweaking the toolpaths is possible, but the affordances these CAM tools have for control are predominately designed around efficiency and not creative exploration. Therefore, experimenting with the aesthetics of different toolpaths is cumbersome and requires some reimagining of the existing tools. The glass artist commented on how she viewed CAM tools as black boxes, where she often needed to hack the CAM setup or even edit the outputted raw G-code in order for her to get to result she wanted. These two observations of the nonlinearity of digital craft workflows and the importance of the marks left by toolpaths are in line with prior studies of people practicing digital craft [Cheatle and Jackson 2015; Twigg-Smith et al. 2021].

Another interesting observation was how the glass artist learned to use our system and machine. By directly interacting with the machine through the Grasshopper canvas, she was able to quickly understand the basic operation of the machine and how she could use it in her different design cases. She was able to reconfigure some of the parameters of the controller to adapt the machines characteristics to be better suited for her designs. Although she did not understand these parameters fully, she was able to gain understanding and utilize them for her needs by interacting with them and observing how they inflicted the machines characteristics as she changed them. This contradicts many existing machine interfaces found on digital fabrication machines where configuration parameters and machine settings are hidden behind high-end interfaces and not made available for users to edit and adapt. This is in line with prior research which finds that while black boxes and simpler controls in tools make tasks more accessible to novices, these forms of automation prevent artists from incorporating these tools in their existing workflows or flexibly extending them across multiple workflows [Li et al. 2021]. We stress that while newcomers to hands-on making are an important user group of digital fabrication, our tools are aimed at practitioners who have existing expertise in fabrication and digital design tools and seek to harness their tacit knowledge of materials and process while extending their workflows to incorporate digital fabrication tools.

We acknowledge several limitations of our contribution. Firstly, while we gleaned many insights from our evaluation, it remains limited to a snapshot of experiments conducted by one artist. While our system supports 3D toolpaths, many of the explorations in our evaluation were 2D cutting or milling patterns. Nonetheless, we believe our evaluation demonstrates that our systems contribution has merit and warrants further exploration. Secondly, we concede that interactive machine control may not be applicable for all types of machines and workflows. We believe interactive control is a great

and valuable feature for smaller sized machines like plotters, 3D printers, laser cutters, or the type of machine that was utilized in our study. However, larger machines like CNC milling machines or industrial robots carry risks for operator safety. They will require a user to be more cautious about the machine programs and movements they are running on the machine, both for the sake of the human operating it and the machine itself. Such tools will require an interaction where a user has to verify the program they have made before executing it.

Finally, our system remains limited to specific machines, namely those which can employ the Duet3D controller. The majority of today’s open-source CNC controllers allow changing and reconfiguring of their properties through variants of G-code. Thus, the system can be extended to support different flavors of G-code and different types of machines controllers. However, different controllers will have different communication interfaces and G-code flavors. To implement a more generalized solution of our system, these subtleties have to be taken into account and mapped out. There is prior work in the literature that proposes schemes for doing this. Adding additional controller support, e.g., for popular machines such as the Axidraw, would be valuable future work. Furthermore, our system is limited to the Grasshopper environment. Grasshopper is one of several CAD tools that is popular in our target group. Future work could develop a more generalized hook that allows makers to incorporate our system into other CAD environments.

By enabling toolpath exploration and direct control within a CAD environment our system allows for exploration an area of digital fabrication that is under-explored. Namely, beyond efficiency and optimization, how do elements of manufacturing such as toolpaths influence the material and aesthetic properties of the fabricated object? In the future, we hope to contribute to many more systems that provide agency to people navigating trade-offs between automation, planning, creativity, and control.

7 CONCLUSION

We believe that craft practices of exploration and iteration produce valuable insights that are difficult to glean when operating in problem-solving mode. As the profile of people using automation evolves from people mainly seeking efficiency to people creatively exploring the opportunities of precision, we need different ways of interacting with and controlling machines. Our contributions demonstrate the possibilities of introducing new types of interaction using existing digital design authoring tools. We designed a system that allows fabrication experts such as craftspeople with experience in using design tools to interact with a machine directly from their CAD workspace. Our system enables the creative exploration of the outcomes of toolpath parameters such as acceleration and jerk on materials and aesthetics. We minimized the number of steps that a maker has to traverse when transforming virtual representation to physical craft. With this we seek to meet craft practitioners where they are: expert fabricators who are already using digital design tools, but are limited by the workflows that those tools afford. Our evaluation with a professional glass artist

demonstrates that the type of workflow that the system provides extends possibilities for digital fabrication, including for experienced practitioners.

ACKNOWLEDGMENTS

We would like to thank Patricia Sichmanova (www.sichmanova.sk) for her patience and time. The insights gleaned from the workshops together with Patricia were highly valuable for the scope of work presented in this article. Thanks to Jake Read for contributing the Clank machine design to the open-source community. Finally, we would like to thank Jens Dyvik (www.dyvikdesign.com/site) for his insights and clarity in digital fabrication workflows and machine design.

REFERENCES

- G. Adamson. 2021. *Craft: An American History*. Bloomsbury Publishing.
- Rahul Arora, Alec Jacobson, Timothy R. Langlois, Yijiang Huang, Caitlin Mueller, Wojciech Matusik, Ariel Shamir, Karan Singh, and David I. W. Levin. 2019. Volumetric Michell Trusses for Parametric Design & Fabrication. In *Proceedings of the ACM Symposium on Computational Fabrication* (Pittsburgh, Pennsylvania) (SCF '19). Association for Computing Machinery, New York, NY, USA, Article 6, 13 pages. <https://doi.org/10.1145/3328939.3328999>
- Leah Buechley and Hannah Perner-Wilson. 2012. Crafting Technology: Reimagining the Processes, Materials, and Cultures of Electronics. *ACM Trans. Comput.-Hum. Interact.* 19, 3, Article 21 (Oct. 2012), 21 pages. <https://doi.org/10.1145/2362364.2362369>
- Patrick Carrington, Shannon Hosmer, Tom Yeh, Amy Hurst, and Shaun K Kane. 2015. "Like This, But Better": Supporting Novices' Design and Fabrication of 3D Models Using Existing Objects. *iConference 2015 Proceedings* (2015).
- Amy Cheate and Steven J. Jackson. 2015. Digital Entanglements: Craft, Computation and Collaboration in Fine Art Furniture Production. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Vancouver, BC, Canada) (CSCW '15). Association for Computing Machinery, New York, NY, USA, 958–968. <https://doi.org/10.1145/2675133.2675291>
- Dipen Dave, Ashirwad Chowriappa, and Thenkurussi Kesavadas. 2013. Gesture Interface for 3D CAD Modeling using Kinect. *Computer-Aided Design and Applications* 10, 4 (2013), 663–669. <https://doi.org/10.3722/cadaps.2013.663-669> arXiv:<https://www.tandfonline.com/doi/pdf/10.3722/cadaps.2013.663-669>
- Garcia del Castillo. 2018. *Machina*. <https://www.food4rhino.com/en/app/machina>
- Himani Deshpande, Haruki Takahashi, and Jeeun Kim. 2021. *EscapeLoom: Fabricating New Affordances for Hand Weaving*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3411764.3445600>
- Audrey Desjardins and Timea Tihanyi. 2019. ListeningCups: A Case of Data Tacility and Data Stories. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) (DIS '19). Association for Computing Machinery, New York, NY, USA, 147–160. <https://doi.org/10.1145/3322276.3323694>
- Laura Devendorf, Katya Arquilla, Sandra Wirtanen, Allison Anderson, and Steven Frost. 2020. *Craftpeople as Technical Collaborators: Lessons Learned through an Experimental Weaving Residency*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376820>
- Duet3D. 2021. Duet3D Minimum configuration with Raspberry Pi. https://duet3d.dozuki.com/Wiki/Getting_Started_With_Duet_3#Section_Setting_up_the_Raspberry_Pi, accessed July 2021.
- Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: Remixing Physical Objects with Copy and Paste from the Real World. In *Adjunct Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) (UIST '10). Association for Computing Machinery, New York, NY, USA, 381–382. <https://doi.org/10.1145/1866218.1866230>
- Frikk H. Fosssdal, Jens Dyvik, Jakob Anders Nilsson, Jon Nordby, Torbjørn Nordvik Helgesen, Rogardt Helda, and Nadya Peek. 2020. Fabricatable Machines: A Toolkit for Building Digital Fabrication Machines. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Sydney NSW Australia, 2020-02-09). ACM, 411–422. <https://doi.org/10.1145/3374920.3374929>
- Raune Frankjær and Peter Dalsgaard. 2018. Understanding Craft-Based Inquiry in HCI. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) (DIS '18). Association for Computing Machinery, New York, NY, USA, 473–484. <https://doi.org/10.1145/3196709.3196750>
- Mark Fuge, Greg Carmean, Jessica Cornelius, and Ryan Elder. 2015. The MechProcessor: Helping novices design printable mechanisms across different printers. *Journal of Mechanical Design* 137, 11 (2015).
- Neil Gershenfeld. 2015. How to Make Almost Anything. <https://www.foreignaffairs.com/articles/2012-09-27/how-make-almost-anything>
- Bruna Goveia da Rocha, Johannes M. L. van der Kolk, and Kristina Andersen. 2021. *Exquisite Fabrication: Exploring Turn-Taking between Designers and Digital Fabrication Machines*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3411764.3445236>
- Jianzhe Gu, Vidya Narayanan, Guanyun Wang, Danli Luo, Harshika Jain, Kexin Lu, Fang Qin, Sijia Wang, James McCann, and Lining Yao. 2020. Inverse Design Tool for Asymmetrical Self-Rising Surfaces with Color Texture. In *Symposium on Computational Fabrication* (Virtual Event, USA) (SCF '20). Association for Computing Machinery, New York, NY, USA, Article 14, 12 pages. <https://doi.org/10.1145/3424630.3425420>
- Megan Hofmann, Gabriella Hann, Scott E. Hudson, and Jennifer Mankoff. 2018. *Greater than the Sum of Its PARTs: Expressing and Reusing Design Intent in 3D Models*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173875>
- Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. *Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers*. Association for Computing Machinery, New York, NY, USA, 384–396. <https://doi.org/10.1145/2858036.2858266>
- Jennifer Jacobs, Sumit Gogia, Radomir Mundefinedch, and Joel R. Brandt. 2017. *Supporting Expressive Procedural Art Creation through Direct Manipulation*. Association for Computing Machinery, New York, NY, USA, 6330–6341. <https://doi.org/10.1145/3025453.3025927>
- Jennifer Jacobs and Amit Zoran. 2015. *Hybrid Practice in the Kalahari: Design Collaboration through Digital Tools and Hunter-Gatherer Craft*. Association for Computing Machinery, New York, NY, USA, 619–628. <https://doi.org/10.1145/2702123.2702362>
- Jeeun Kim, Haruki Takahashi, Homei Miyashita, Michelle Annett, and Tom Yeh. 2017. Machines as Co-Designers: A Fiction on the Future of Human-Fabrication Machine Interaction. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI EA '17). Association for Computing Machinery, New York, NY, USA, 790–805. <https://doi.org/10.1145/3027063.3052763>
- Maria Larsson, Hironori Yoshida, and Takeo Igarashi. 2019. Human-in-the-Loop Fabrication of 3D Surfaces with Natural Tree Branches. In *Proceedings of the ACM Symposium on Computational Fabrication* (Pittsburgh, Pennsylvania) (SCF '19). Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. <https://doi.org/10.1145/3328939.3329000>
- Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. *Tsugite: Interactive Design and Fabrication of Wood Joints*. Association for Computing Machinery, New York, NY, USA, 317–327. <https://doi.org/10.1145/3379337.3415899>
- Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. *JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300386>
- Jingyi Li, Sonia Hashim, and Jennifer Jacobs. 2021. *What We Can Learn From Visual Artists About Software Development*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3411764.3445682>
- Jingyi Li, Jennifer Jacobs, Michelle Chang, and Björn Hartmann. 2017. Direct and Immediate Drawing with CNC Machines. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication* (Cambridge, Massachusetts) (SCF '17). Association for Computing Machinery, New York, NY, USA, Article 11, 2 pages. <https://doi.org/10.1145/3083157.3096344>
- Zhao Ma, Simon Duenser, Christian Schumacher, Romana Rust, Moritz Bächer, Fabio Gramazio, Matthias Kohler, and Stelian Coros. 2020. RobotSculptor: Artist-Directed Robotic Sculpting of Clay. In *Symposium on Computational Fabrication* (Virtual Event, USA) (SCF '20). Association for Computing Machinery, New York, NY, USA, Article 13, 12 pages. <https://doi.org/10.1145/3424630.3425415>
- Shiran Magrisso, Moran Mizrahi, and Amit Zoran. 2018. *Digital Joinery For Hybrid Carpentry*. Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3173574.3173741>
- Robert McNeel and Associates. 2021. Rhino & Grasshopper component development in C#. <https://developer.rhino3d.com/guides/grasshopper>, accessed July 2021.
- Evgeny Morozov. 2014. Making it. <https://www.newyorker.com/magazine/2014/01/13/making-it-2>.
- Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive Construction: Interactive Fabrication of Functional Mechanical Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (UIST '12). Association for Computing Machinery, New York, NY, USA, 599–606. <https://doi.org/10.1145/2380116.2380191>
- Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann. 2018. Automatic Machine Knitting of 3D Meshes. *ACM Trans. Graph.* 37, 3, Article 35 (Aug. 2018), 15 pages. <https://doi.org/10.1145/3186265>
- Martin Nisser, Christina Chen Liao, Yuchen Chai, Aradhana Adhikari, Steve Hodges, and Stefanie Mueller. 2021. *LaserFactory: A Laser Cutter-Based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3411764.3445692>
- Andy Payne and Jason Kelly Johnson. 2010. *Firefly*. <https://www.food4rhino.com/en/app/firefly>

- Nadya Peek, James Coleman, Ilan Moyer, and Neil Gershenfeld. 2017. Cardboard Machine Kit: Modules for the Rapid Prototyping of Rapid Prototyping Machines. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA) (CHI '17). ACM, 3657–3668. <https://doi.org/10.1145/3025453.3025491>
- Nadya Peek and Ilan Moyer. 2017. Popfab: A Case for Portable Digital Fabrication. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '17*. ACM Press, Yokohama, Japan, 325–329. <https://doi.org/10.1145/3024969.3025009>
- Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. 2015. *A Layered Fabric 3D Printer for Soft Interactive Objects*. Association for Computing Machinery, New York, NY, USA, 1789–1798. <https://doi.org/10.1145/2702123.2702327>
- Jie Qi, Andrew "bunnie" Huang, and Joseph Paradiso. 2015. Crafting Technology with Circuit Stickers. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) (IDC '15). Association for Computing Machinery, New York, NY, USA, 438–441. <https://doi.org/10.1145/2771839.2771873>
- Jake Read. 2021. Clank: straightforward, easy to fabricate and assemble machine platforms. <https://gitlab.cba.mit.edu/jakeread/clank>, accessed July 2021.
- Michael L. Rivera and Scott E. Hudson. 2019. *Desktop Electrospinning: A Single Extruder 3D Printer for Producing Rigid Plastic and Electrospun Textiles*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300434>
- Alec Rivers, Ilan E. Moyer, and Frédo Durand. 2012. Position-correcting tools for 2D digital fabrication. *ACM Transactions on Graphics* 31, 4 (July 2012), 1–7. <https://doi.org/10.1145/2185520.2185584>
- Daniela K. Rosner, Samantha Shorey, Brock R. Craft, and Helen Remick. 2018. *Making Core Memory: Design Inquiry into Gendered Legacies of Engineering and Craftwork*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3174105>
- Thijs Jan Roumen, Willi Müller, and Patrick Baudisch. 2018. *Grafter: Remixing 3D-Printed Machines*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173637>
- Ryan Schmidt and Matt Ratto. 2013. Design-to-Fabricate: Maker Hardware Requires Maker Software. *IEEE Computer Graphics and Applications* 33, 6 (2013), 26–34. <https://doi.org/10.1109/MCG.2013.90>
- Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. Drill Sergeant: Supporting Physical Construction Projects through an Ecosystem of Augmented Tools. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) (CHI EA '16). Association for Computing Machinery, New York, NY, USA, 1607–1614. <https://doi.org/10.1145/2851581.2892429>
- Behrooz Tahanzadeh. 2015. *Bengesht*. <https://www.food4rhino.com/en/app/bengesht>
- Haruki Takahashi and Jeeun Kim. 2019. *3D Pen + 3D Printer: Exploring the Role of Humans and Fabrication Machines in Creative Making*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300525>
- Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 713–725. <https://doi.org/10.1145/3332165.3347918>
- Rundong Tian, Sarah Sterman, Ethan Chiou, Jeremy Warner, and Eric Paulos. 2018. *MatchSticks: Woodworking through Improvisational Digital Fabrication*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173723>
- Cesar Torres, Wilmot Li, and Eric Paulos. 2016. ProxyPrint: Supporting Crafting Practice through Physical Computational Proxies. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (Brisbane, QLD, Australia) (DIS '16). Association for Computing Machinery, New York, NY, USA, 158–169. <https://doi.org/10.1145/2901790.2901828>
- F. Turner. 2010. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. University of Chicago Press.
- Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2021. *Tools, Tricks, and Hacks: Exploring Novel Digital Fabrication Workflows on #PlotterTwitter*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3411764.3445653>
- Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2020. Jubilee: An Extensible Machine for Multi-Tool Fabrication. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376425>
- Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 623–635. <https://doi.org/10.1145/3242587.3242625>
- Guanyun Wang, Lining Yao, Wen Wang, Jifei Ou, Chin-Yi Cheng, and Hiroshi Ishii. 2016. XPrint: A Modularized Liquid Printer for Smart Materials Deposition. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 5743–5752. <https://doi.org/10.1145/2858036.2858281>
- Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2010. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction* (Funchal, Portugal) (TEI '11). Association for Computing Machinery, New York, NY, USA, 69–72. <https://doi.org/10.1145/1935701.1935716>
- Lining Yao, Jifei Ou, Guanyun Wang, Chin-Yi Cheng, Wen Wang, Helene Steiner, and Hiroshi Ishii. 2015. bioPrint: A liquid deposition printing system for natural actuators. *3D Printing and Additive Manufacturing* 2, 4 (2015), 168–179.
- Nur Yildirim, James McCann, and John Zimmerman. 2020. *Digital Fabrication Tools at Work: Probing Professionals' Current Needs and Desired Futures*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376621>
- Amit Zoran and Joseph A. Paradiso. 2013. FreeD: a freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, Paris, France, 2613. <https://doi.org/10.1145/2470654.2481361>