Lucid Fabrication

Rundong Tian

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-165
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-165.html

July 9, 2021

Lucid Fabrication

by

Rundong Tian

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Eric Paulos, Chair
Professor Björn Hartmann
Professor Simon Schleicher
Professor Nadya Peek

Summer 2021

Lucid Fabrication

Abstract

Lucid Fabrication

by

Rundong Tian

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Eric Paulos, Chair

The growing availability of digital fabrication tools has transformed how diverse communities of makers design, prototype, and manufacture physical objects. However, in order to fluently create physical objects using these tools, users must navigate a gauntlet of software packages to draft a digital model, devise a toolpath for manufacturing, and drive the fabrication machine itself. In my dissertation research, I consider alternative workflows and interactions that allow users to more fluidly engage with the capabilities of digital fabrication tools. My exploration in this domain is guided by the following question: *How can digital fabrication tools engage and amplify opportunities for human judgment, skill, and creativity during the fabrication process?*

This dissertation engages with this question from three perspectives embodied in three respective interactive systems. First, how can design intent be directly communicated to a digital fabrication tool? I examine this through MatchSticks, a bespoke CNC system that localizes design and fabrication workflows, and allows users to rapidly design and create wood joinery in-situ. Second, how can the existing ways users interact with physical tools be augmented, rather than defenestrated, by computation? Using a manual lathe as a case study, I exchange the mechanical coupling between handwheels and tool position for a digital coupling: sensors, actuators, computation. Users directly control this lathe "by-wire", while being supported by capabilities more often associated with digital fabrication. Last, how can the capabilities we associate with digital authoring be broadly incorporated within hands-on making? I examine how commonly used workshop jigs and fixtures can be generated computationally using an industrial robot. The resulting interactions afford the tangible familiarity of physical jigs and fixtures while taking full advantage of reprogrammable software.

All three approaches acknowledge and celebrate the process of fabrication as a site of creative exploration and problem solving. With this framing, I demonstrate how not only can capabilities of digital fabrication be realized outside of established workflows, but that hands-on fabrication itself can be imbued with the characteristics of digital authoring.

To my family: Mom, Dad, Jackie, and Moomar.

Contents

List of Figures

Acknowledgments

To my advisor, Eric Paulos. My HCI journey began when you welcomed me into your lab in 2013, when I was an undergraduate at Berkeley trying to figure out what research was all about. Thank you for your trust and guidance through the many explorations that eventually led to this dissertation. The Hybrid Ecologies Lab is truly a one of a kind research oasis. Thank you for creating and nurturing such a welcoming, creative, and supportive environment.

I am fortunate to have had the counsel and support of my dissertation committee. To Björn Hartmann, thank you for your insights and feedback throughout this journey. To Simon Schleicher, your architecture seminars honed my research skills and gave me the space to explore many new ideas. To Nadya Peek, the scope of hardware projects that you created during your PhD has been an inspiration. Thank you as well for introducing me to the history of digital fabrication.

To my colleagues in the Hybrid Ecologies Lab and beyond: Cesar Torres, Sarah Sterman, Jeremy Warner, Tim Campbell, Joanne Lo, Molly Nicholas, Christie Dierk, Katherine Song, Vedant Saran, Laura Devendorf, Noura Howell, and Chris Myers. Your works, advice, and insights have guided my perspectives on digital fabrication and shaped my approach to research.

To my research mentors at Siemens, Mareike Kritzler and Florian Michahelles. Thank you for allowing me to join the Siemens research outpost in downtown Berkeley, and for your support as we navigated the challenges of *Turn-by-Wire*.

To my undergraduates throughout the years: Ethan Chiou, Tomás Vega, Eric O'neill, John Kuwata, and Juan Carrillo. Thank you all for your support on the projects within and beyond this dissertation.

To my best friend and now wife, Jackie Berry. I am so lucky to have met you all those years ago at nerd camp, and I am blessed for your love and patience. A lifetime of adventures awaits!

To mom and dad. You all have made it possible for me to explore my interests, wherever they led. Without your love and support, this moment would not only have been impossible, but unimaginable.

Chapter 1

Introduction

With the aid of sensors and actuators, the motion of any tool—such as a plastic extruder, laser cutter, or spinning end mill—can be controlled through computation. This arrangement, called digital fabrication, allows raw materials to be accurately and autonomously shaped into complex geometries. These types of tools where once only available in professional settings, owing in part to the prohibitive cost the tools and the software required to use them. Beginning in the early 2000s however, a constellation of factors began to erode the cost and technical barriers of digital fabrication. These factors include the expiration of key patents, the explosion of affordable desktop fabrication machines, the rise of on-demand fabrication services, the availability of open-source hardware and software, as well as the growth of online and in-person communities around making [3, 10]. Digital fabrication tools and their surrounding communities has been one pillar of the broader maker movement, which emphasizes creative engagement through making—whether it be food, electronics, or furniture [28].

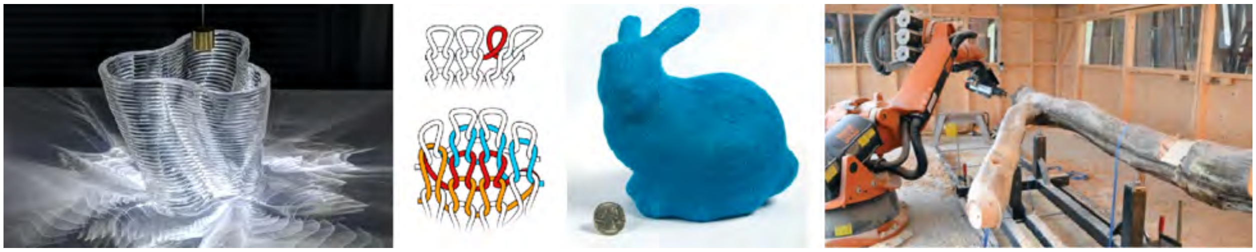


Figure 1.1: Fabricating with glass [53], textiles [80], and tree-trunks [24].

The growing availability of digital fabrication tools has transformed the process of physical making for engineers, designers, artists, and dilettantes alike [73]. Not only that, digital fabrication encompasses an ever growing list of capabilities: sculptures can be 3D printed from coils of molten glass [53], timberframes can be machined with robotic arms [24], and bespoke sweaters can be automatically knit using computationally generated patterns [80]. However, this breadth of applications should not imply that these tools are a blank canvas for creative expression. As with all tools, the capabilities of a digital fabrication machine defines how objects are fabricated, its constraints regulate what kinds of objects are designed, and its surrounding workflows shape how and by whom objects are conceived and created.

Specifically, this dissertation seeks to address the challenges of working with conventional digital fabrication workflows. Suppose a user is designing a plywood chair, and intends to fabricate it using a computer guided milling machine. With existing workflows, she must draft a digital model of the object using a Computer Aided Design (CAD) software, define how the milling tool should move with a Computer Aided Manufacturing (CAM) software, and set up the milling machine to execute the fabrication instructions using a user interface specific to that machine's manufacturer. The machine mindlessly follows where the pre-calculated fabrication instructions command it to go. Meanwhile, the user has little authority over the motions of the tool, other than adjusting the overall speed, or stopping the tool entirely. If there are any errors—for example, an unaccounted for difference between the real and simulated environment that causes the mill to crash—the user often cannot correct it at the machine, but must retrace her steps through the software pipeline. In sum: the process is viscous, and the outcome is brittle.

The hypothetical user in the last paragraph might be a college student working on a design assignment in a makerspace, an artist creating a small run of identical chairs in their studio [19], or a hobbyist creating custom furniture in their garage. In a professional setting, the hypothetical user might actually be many users, each of whom is responsible for one stage of the pipeline—a designer creates the design, the CAM programmer creates the toolpath, and a technician operates the machine.

Depending on the context, the challenges of working with conventional workflows might vary in significance. One simple but important way in which these contexts differ is the number of times that objects are replicated—the student might make the same object only once, the artist studio might recreate the same chair a dozen times, and the professional woodworking shop might regularly fabricate the same component hundreds of times, if not more. If the same object is fabricated repeatedly, the initial time and effort required to traverse this workflow is amortized, and the fabrication environment can be tightly controlled to reduce potential errors. The benefits of this workflow are highlighted as well: the components are fabricated without user intervention, and machining parameters can be optimized for speed. Surprisingly however, the conventional workflow is approximately the same whether a user intends to make one million of the same object, or a single object. More surprising still is the fact that while the capabilities of digital fabrication tools facilitate mass customization, its canonical workflow favors mass production.

The discourse around digital fabrication has sometimes orbited the assumption that with with access to digital fabrication equipment, anyone can make nearly anything [39]. "Anyone" in this case implicitly means: anyone who is, or wishes to become, fluent in the suite of design and manufacturing software required to use these machines. These software pipelines are functional and mature, but they generate enormous overhead for communicating design intent to a fabrication tool. Even after a user become fluent in these tools, they may find that the workflows are poorly suited for the type and volume of objects they want to create. How might alternative workflows enable users to interact with the capabilities of digital fabrication tools more fluidly?

This dissertation seeks to address these challenges by first acknowledging that the process of fabrication can be a site of creative exploration and problem solving. Rather than further removing humans from the process of physical fabrication, my exploration in this domain is guided primarily by the following research question: *How can digital fabrication tools engage and amplify opportunities for human judgment, skill, and creativity during the fabrication process?* Through three interactive systems guided by this framing, I demonstrate how not only can the properties we associate with digital fabrication be achieved outside of traditional Computer Aided Design and Manufacturing workflows, but that physical hands-on fabrication itself can be imbued with the characteristics of digital editing.

1.1 Contribution summary

The primary contribution of this dissertation is the design and engineering of three interactive systems. Each of these systems embody new ideas and opportunities for how we might interact with digital fabrication tools, as well the techniques for how these opportunities can be realized. We introduce each project below.

Figure 1.2: *MatchSticks* [120], *Turn-by-Wire* [121], and *aDroid*.

MatchSticks is a digital fabrication system tailored for woodworking joinery that enables makers to creatively explore and rapidly create artifacts from wood. This system consists of a portable computer controlled router, touch screen user interface, and parametric joint library. It distills existing distributed software workflows to the site of the machine itself, operates on materials existing machines find difficult, and produces assemblies much larger than the tool itself. I present artifacts produced by this tool and report on results from a user study.

Turn-by-Wire explores how computation can augment the manual control of fabrication tools. Using a manual lathe as a case study, I exchange the mechanical handwheels with digitally reprogrammable ones—haptic controllers which allow the user to command the position of the lathe, as well as the lathe to communicate information to the user. Users control the lathe directly and manually, but “by-wire”. This digitally mediated interaction allows for many of digital fabrication capabilities (automatic accuracy, complex geometries, autonomous duplication) to be directly accessible through manual control. I report on the results of a user study with expert fabricators from a broad range of backgrounds.

aDroid uses an industrial robot to computationally generate and augment physical jigs and fixtures. When a tool is mounted to the robot, the user holds and moves the tool directly, and backdrivability is achieved through force-sensing and software. Users can borrow precision and accuracy from the robot as needed by adapting the backdrivability—the virtual jig—of the robot for each task. These interactions are complemented by a projection augmented reality display for in-situ visual feedback about the state of the system. I demonstrate the generalizability of this approach with four tools, each of which showcases a unique facet of the system. In addition, to begin to address the prohibitive cost of robots and their accessories, I detail the design and implementation of a low cost force-torque sensor which is central to the types of interaction presented.

1.2 Structure

The structure of this dissertation is as follows:

Background and Related Work

In the last ten years, research in digital fabrication has burgeoned in the the Computer Graphics and Human Computer Interaction (HCI) communities [10]. Within this growing landscape, this dissertation focuses on the physical fabrication tools themselves and the ways in which users interact with them. This chapter will organize and discuss different classes of fabrication tools, the relationships between them, and the broader workflows that they are a part of. To further contextualize the field, I will discuss technologies proposed during the dawn of numerically controlled tools in the mid-twentieth century, as well as research within the adjacent domains of manufacturing and architecture.

Three Perspectives: *MatchSticks*, *Turn-by-Wire*, *aDroid*

This dissertation explores the domain of digital fabrication through a series of projects embodying reciprocal concepts. I first present *MatchSticks*, a tool for joinery, which combines a context specific computer controlled machine with turn-taking interactions that allow the user to fluidly specify the type of joint they want to create. This combination of machine and interactions allows users to rapidly prototype functional and aesthetic wooden objects, without ever having to navigate CAD/CAM workflows in their entirety.

Turn-by-Wire inverts many of the concepts guiding the design of the *MatchSticks* system — instead of directly communicating design intent, it explores programmable haptic feedback; instead of graphical user interface (GUI) centric interactions, it focuses on foregrounding and augmenting existing manual ways of working with machines. The force feedback enabled by *Turn-by-Wire*'s haptic handwheels is not intended to replicate the sensation of using a real manual lathe. Instead, when combined with "by–wire" control, capabilities more often associated with digital editing — such as snap-to-grid, or in context guidance that teach through touch — can be incorporated within hands–on fabrication.

To generalize the ideas explored by *Turn-by-Wire*, the last project considers how computation can mediate the entire making environment, rather than individual tools. By leveraging the concept of virtual jigs and fixtures, *aDroid* allows users to work directly with physical tools, while borrowing from the accuracy, strength, and programmability of an industrial robot. Using this architecture, I begin to incorporate capabilities of digital authoring directly at the site of hands-on making. I demonstrate the efficacy of this system in supporting multiple tools and discuss how this system can be rapidly reconfigured to support a variety of workshop fabrication tasks.

Discussion and conclusion

After presenting the three projects, I discuss their overall approach, trade-offs, and limitations. I conclude with a discussion of future work of how these concepts could be broadly applied to digital fabrication, manufacturing, and beyond.

1.3 Statement of multiple authorship and prior publication

This dissertation draws upon work previously published at the ACM UIST 2019 and ACM CHI 2018 conferences (*Turn-by-Wire* [121] and *MatchSticks* [120], respectively). Although I led the research and engineering efforts of each project, these works were inspired, informed, and shaped by the members of the Hybrid Ecologies Lab—Cesar Torres, Tim Campbell, Joanne Lo, Laura Devendorf, Christie Dierk, Sarah Sterman, Molly Nicholas, Vedant Saran, Katherine Song, Chris Myers, and Kuan-Ju Wu. For Matchsticks, early prototypes of the user interface where created by Ethan Chiou and Eric Liang. User studies were conducted and analyzed with Sarah Sterman, and the tutorial workflow user interface was created by Jeremy Warner. For *Turn-by-Wire*, Vedant Saran developed AR prototypes and helped conduct user studies. The freedom to begin exploring this concept in depth was generously granted to me by my advisors at Siemens Research, Florian Michahelles and Mareike Kritzler. My advisor, Eric Paulos, has provided invaluable guidance and insight on all projects presented in this dissertation.

Chapter 2

Background and related work

This chapter centers the discussion of digital fabrication on the fabrication tools themselves: how users interact with them, what workflows they are a part of, and the compatibility between different types of tools and interactions. Three categories of related work will be discussed in further detail. To situate *Turn-by-Wire* and *aDroid*, I describe other works which allow users to directly and continuously control the fabrication process, and organize these works based on how and why these systems incorporate computation (Section 2.3). To contextualize *MatchSticks*, I discuss related strategies which compress conventional digital fabrication workflows (Section 2.4). Lastly, I introduce research in Computer Aided Design in Section 2.5, and begin to unpack the various ways in which computers can "aid" the design process upstream of the fabrication tools themselves.

2.1 Methods for controlling fabrication tools

Digital fabrication tools allow makers to shape, join, extrude, cut, melt, or otherwise modify physical materials. In this section, we will discuss the conventional workflow for engaging with digital fabrication tools, as well as other methods that embody significantly different strategies.

2.1.1 Computer numeric control

Computer numeric controlled tools operate by executing *toolpaths*: a list of instructions that specify how an object should be fabricated in terms of where the tool should move, how fast it should move, as well as other parameters. Many CNC tools run on a type of instructions called G-code, which has its own syntax for specifying these types of instructions. For example, the command "`G1 X10 Y20 Z30`" will move the machine in straight line from its current position to the point (10, 20, 30) in the machine's current coordinate system.

While it is certainly possible for a CNC machinist to directly write G-code in a text editor, this process would be akin to programming in assembly. Rather, a more likely workflow is to "compile" the desired G-code instructions using existing software workflows. First, using a Computer Aided Design (CAD) software, the user can build a digital representation of the desired geometry. Next, this geometry is passed along to a Computer Aided Manufacturing (CAM) software, in which the user specifies how they wish the for the component to be autonomously fabricated. For subtractive manufacturing, this would entail selecting the appropriate tools, cutting strategies, and cutting rates; for additive manufacturing, slicing parameters. While CAD and CAM can exist as entirely separate software packages, companies such as Autodesk have unified them within the same user interface (Figure 2.1).

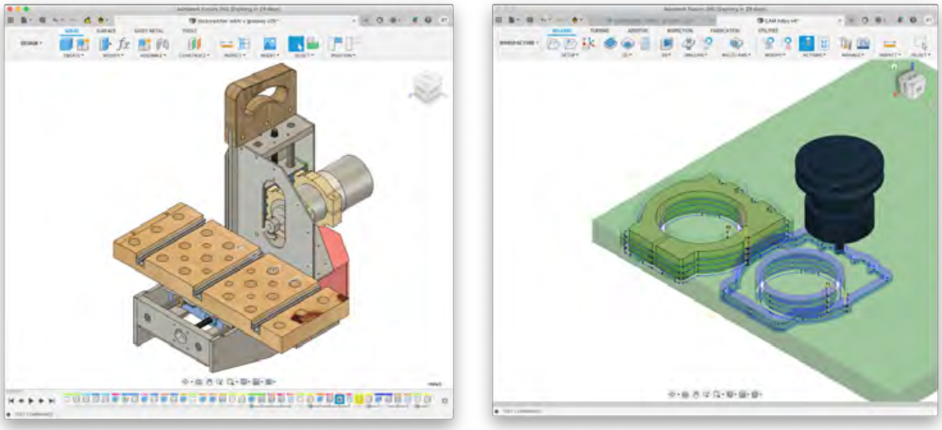


Figure 2.1: Modern design software, such as Autodesk Fusion 360, often contain both CAD and CAM functionalities within the same environment. Left: screenshot of a CAD model of the MatchSticks device. Right: CNC toolpath for machining the router mount.

CAD/CAM/CNC workflows can introduce significant overhead. A user must create a digital representation of the desired geometry, translate that geometry into specifications for how a fabrication machine should move, then operate the setup of the machine through yet another control software. Using this workflow, as well as learning how to use this workflow, can be a viscous process.

These characteristics of this conventional workflow has changed little since its inception. Figure 2.2 describes the process for creating a part with the APT system, a precursor to modern CNC workflows [101]: a designer conceives of a geometry, a draftsman translates the geometry into a drawing, a programmer translates the drawing into a programming language that specifies the movement of the tool, and the APT computer compiles that program into punched tape. The machinist, hands in his pockets, is relegated to watching the machine be controlled by the punched tape. After detailing the many stages of this workflow, which involves at least six different individuals, the final panel of Figure 2.2 unironically declares, "And the part is automatically made." The only role that APT seems to have "automated" is that of the machinist. To do so, this system transferred control over how to machine the component—devising the appropriate fixturing, tooling, and motions—from the machinist to

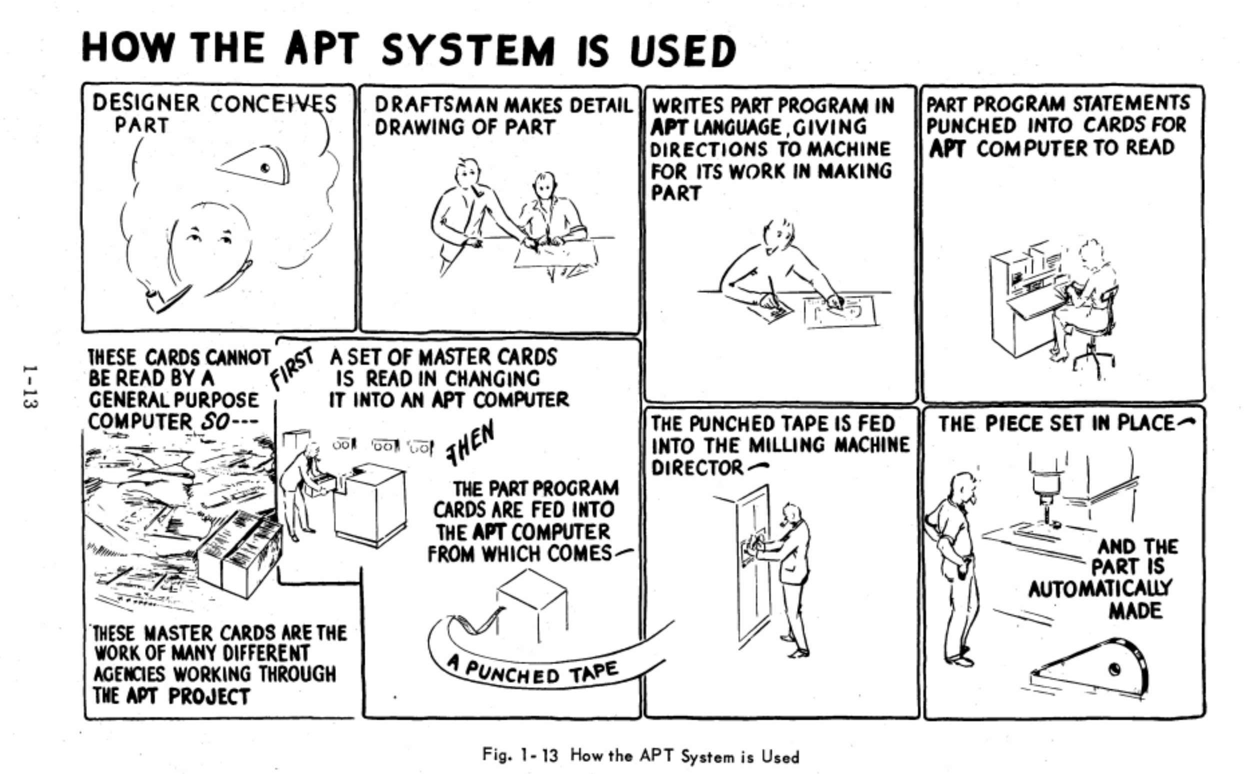


Figure 2.2: Working with the Automatically Programmed Tool system (APT). Retrieved from *General Description of the APT System*, 1959 [101].

the programmer. In modern CNC workflows, the sequence of translations between different representations of the desired geometry occurs exclusively within software; simultaneously, it is possible for a single user to navigate all of these translations. However, the overall process for creating an object has largely remained the same.

2.1.2 Continuous embodied control

One of the earliest technologies for digital fabrication sidestepped many of the complexities of the CAD/CAM/CNC workflow by simply allowing the machinist to maintain control over the fabrication process. Aptly named *record-playback*, it allowed the machinist to define a toolpath by simply manually machining the first part [81]. The way that a machinist moved the tool was recorded by the system, and could be replayed to create subsequent components (Figure 2.3). Rather than create a method which abstracted and reconfigure who programmed the machine, *record-playback* built upon existing practices—the machinist can program the tool simply by working with it the way they normally would.

Record-playback demonstrated overlapping functionality as numeric control, and did so

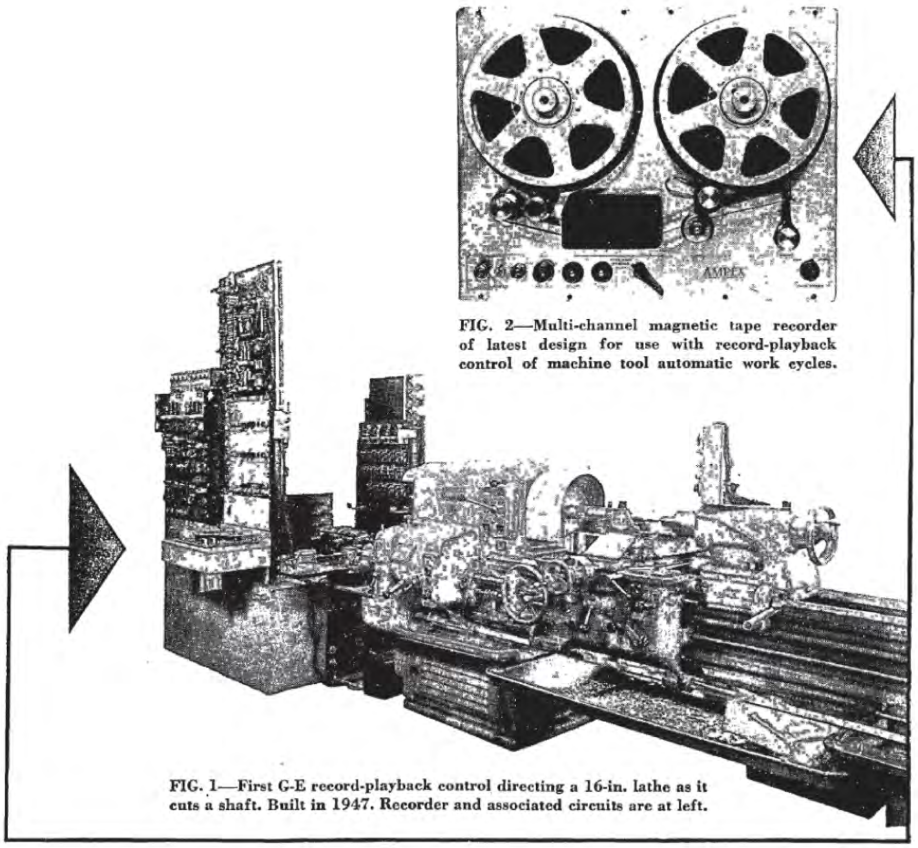


FIG. 2—Multi-channel magnetic tape recorder of latest design for use with record-playback control of machine tool automatic work cycles.

FIG. 1—First G-E record-playback control directing a 16-in. lathe as it cuts a shaft. Built in 1947. Recorder and associated circuits are at left.

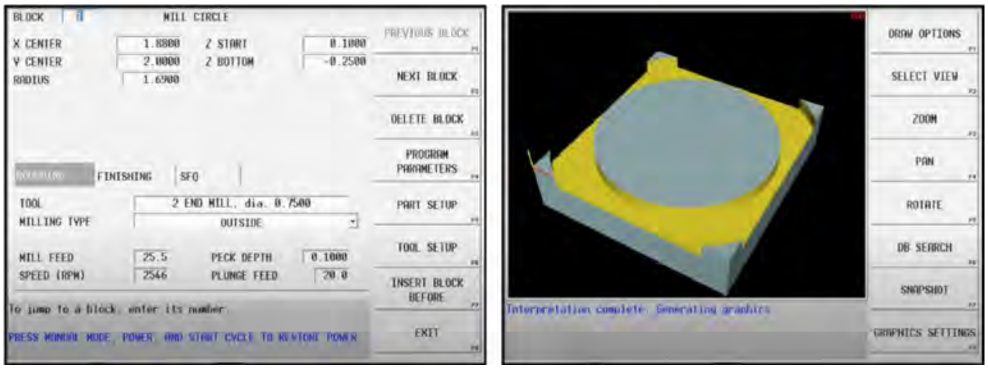
Figure 2.3: GE's *record-playback* system, Retrieved from *Electrical Manufacturing*, November 1953 [84]. Found through *Forces of Production*, page 238 [81].

using simpler techniques. Considering these similar technologies which were developed at around the same time, how did numeric control become the predominant way in which digital fabrication tools are controlled? In *Forces of Production*, David Noble argues that numeric control, unlike its competitors, was developed under the financial auspices of the United States military, and was enthusiastically adopted by corporate management seeking to wrest control of production away from the machinists at the factory floor [81].

To program a toolpath that can be executed automatically, GE's *record-playback* lathe first needed to be continuously controlled by a skilled machinist, much like how a manual lathe is controlled. We term this style of control *continuous embodied control*. Computation was not utilized to remove the need for human input entirely, but to amplify it in a simple way through automatic replication. Many works in HCI today—such as *Interactive Fabrication* [136], hybrid tools [146], as well as the works in this dissertation—continue this idea of augmenting the process of hands-on making. In section 2.3, we detail various ways in which researchers have combined computation and hands-on making.

2.1.3 Conversational control

Conversational control similarly considered how programming and machining could be reconsolidated. With this style of control, machinists could create toolpaths for the fabrication tool directly at the machine. Through an on-tool interface, the machinist can program commonly used operations such as drilling a hole pattern or machining a pocket. Hurco, a machine tool manufacturer, claims to have invented this method in the 1970s [21]. This style of control trades a lower floor for narrower walls: not necessarily all geometries can be articulated using conversational control, but for the types of geometries that can be, the process of generating fabrication instructions can potentially be much simpler. In effect, conversational control makes a compressed version of the CAD/CAM pipeline available at the machine itself. Today, nearly all machine tool manufacturers have incorporated this kind of control within their machines' interfaces [22, 70]. HCI research, such as *Interactive Construction* [75] and my own work, *MatchSticks* [120], has also applied similar ideas to desktop fabrication tools.



X CENTER 1.8800
Y CENTER 2.0000
RADIUS 1.6900
Z START 0.1000
Z BOTTOM -0.2500
TOOL 2 END MILL, dia. 0.7500
MILLING TYPE OUTSIDE
MILL FEED 25.5
SPEED (RPM) 2546
PECK DEPTH 0.1000
PLUNGE FEED 20.0

To jump to a block, enter its number.

PRESS MANUAL MODE, POWER, AND START CYCLE TO RESTORE POWER

Interpretation complete. Generating graphics

Figure 2.4: With conversational machining, users can define geometries to cut, and how to cut them, using a GUI wizard located at the machine. Screenshots retrieved from Hurco CNC training material [83].

In the original patent, this technique was referred to as the *Interactive Machining System*, and sought to create a *"means for programming part features by the machine operator as the machine location"* [98]. Contrasting with more recent projects such as *Interactive Fabrication* [136], these earlier works share overlapping motivation as well as language. In section 2.4, we will explore works that similarly seek to shorten the process of creating fabrication instructions.

2.1.4 STEP Numeric Control

The instructions produced by CAD/CAM workflows, or by conversational control, can be extremely brittle. The toolpath has no understanding of design intent. Rather than encoding information about a component's geometry, its required tolerances, or desired surface finish, G-code primarily consists of a list of cartesian positions for the fabrication machine to move to. The machine will blindly actuate to these positions, regardless of discrepancies between reality and the digital environment that the toolpath was created in. In other words, having a properly fabricated component is only a byproduct of a perfect setup, not something that can be directly specified.

STEP-NC, a new international standard yet to be widely adopted, aims to replace G-code by allowing STEP (a common boundary representation geometry format), the geometry itself, to be an input to a digital fabrication tool [138]. Technically, this can be viewed as simply shifting the computation for creating toolpaths away from CAM software packages, to the fabrication machines themselves. However, the machine itself can have significantly more information about what toolpath strategies would be appropriate, and introduces opportunities for closed loop feedback. More importantly, this approach allows a user to communicate a desired geometry to the fabrication tool directly, rather than incidentally.

2.2 Untangling workflows

Figure 2.5 presents a simplified view of the many paths a user might navigate as they move from an idea to a fabricated object. This representation trades clarity for resolution. For example, less common workflows (such as writing G-code in a text editor) have been excluded. STEP-NC may require something akin to CAM, but conceptually, it controls the operation of a tool based on a desired geometry rather than a toolpath. In addition, this diagram presents a strictly linear view of the fabrication process, and does not try to capture how users might use these tools to iterate on a design. However, through this simplification, we can discuss the types of workflows that are compatible with different classes of tools.

As was previously discussed, the pipeline leading to computer numeric control can be one of the most involved, in terms of the number of times that a user has to translate between one representation of the desired geometry to another. On the other hand, while continuous embodied control and conversational control embody radically different ways of interacting with a fabrication tool, users can engage with these tools much more flexibly. For tools in both of those categories, a user can approach the tool with an idea, and start working with the tool toward a desired geometry immediately. If more planning is desired, a user can develop the geometry further through sketches before committing to fabrication. Using CAD to create and manage a digital representation first is also possible, but literacy with these techniques is not assumed nor required. Tools which support these more flexible workflows are the focus of this dissertation—*MatchSticks* resides within Conversational Control, while *Turn-by-Wire* and *aDroid* reside within Continuous Embodied Control. This examination through the lens of workflows is similar to recent work by Twigg-Smith et al., which analyzed the tools and workflows developed by an online community of computer-controlled drawing machine users—`#PlotterTwitter` [126]. The authors found that this community developed and shared tools that deviated dramatically from the canonical CAD/CAM/CNC workflow—oftentimes as a direct response to the limitations imposed by that workflow—and calls upon

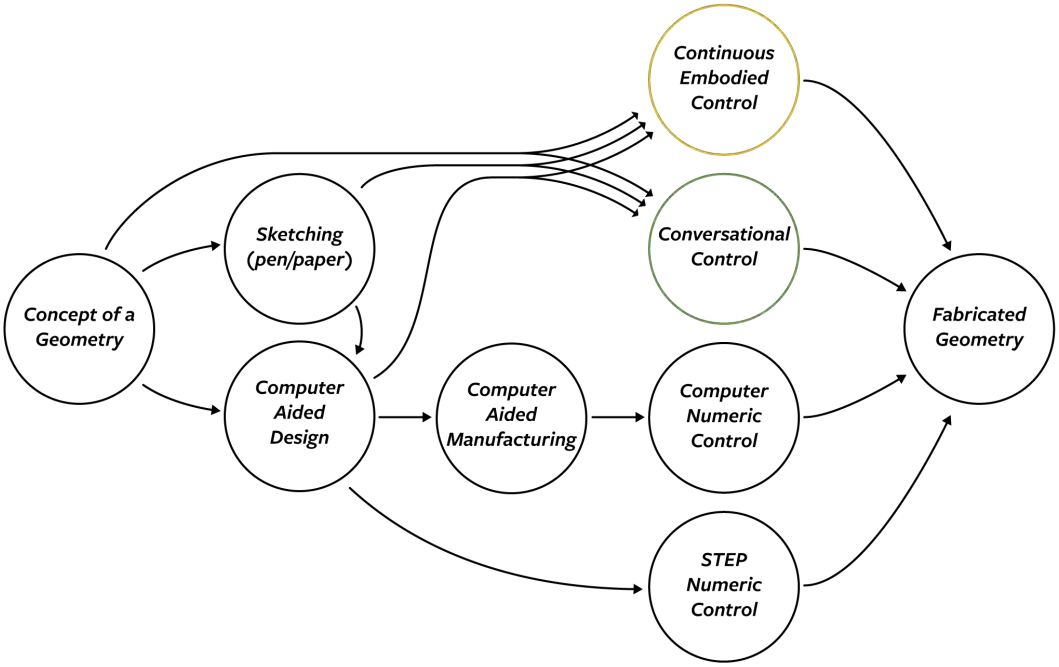


Figure 2.5: A maker can take many potential paths to go from the idea of a geometry, to the final fabricated form. The projects in this dissertation reside within the orange (*Turn-by-Wire* and *aDroid*) and green (*MatchSticks*) circles.

researchers to develop digital fabrication tools which afford flexibility over standardization. This dissertation is similarly motivated by a desire to broaden the ways in which users can engage with the capabilities of digital fabrication.

One principle difference between how fabrication tools are controlled is whether their motions are human or machine operated. Many factors—regarding the geometry, material, or tool—can influence which type of control is appropriate for a particular fabrication task. For example, What is the complexity of the overall geometry? What is the desired resolution and accuracy? How many copies of a component are required? Will the geometry be fabricated using a precious material, or a replaceable one? Is the process resilient to errors, or sensitive to errors? What is the cost of an error? Does fabricating one part take a few minutes or a few hours?

The more a fabrication task lies to the right of these spectra, the more likely it is that automatic control (CNC and Conversational) is more appropriate. For example, a 3D printer can make complex geometries out of replaceable materials, and the printing process can be tightly monitored and controlled by computation. Even if an error does occur during the print, the object can can be reprinted, with the only cost being the machine's time and material. In addition, 3D printing often occurs at the time scale of hours, even for small parts. Especially because of how slow the process is, continuous control can be much less

Figure 2.6: Several of the factors that come into play when considering which style of control is suitable for a particular fabrication task.

desirable than developing a system which can automatically fabricate the desired geometry.

Conversely, continuous embodied control becomes more applicable for tasks that lie towards the left of these spectra. For example, a subtractive fabrication tool like a lathe can shape material much more rapidly than a 3D printer. Because of the higher forces involved in this process, the cost of an error can be far greater — at best, the tool breaks or the component is scrapped, at worst, the machine needs to be recalibrated or rebuilt by a technician. These cost of an error are further amplified if the user is working with precious materials or tools. Especially when creating a few of the same components, continuous control becomes more desirable. Users directly control the machine to rapidly fabricate geometries, while avoiding an entire class of errors that are unique to automatic control—those that arise if the real environment that the toolpath is executed in deviates from the simulated environment a toolpath is created in. However, if many of the same components are desired, the benefits of automatic control may carry more weight.

2.3 Mediating continuous embodied control

Computation can intersect continuous embodied control in numerous ways. It is important to note that even simple tools utilize computation—a modern electric router is likely controlled by a microcontroller that (at a minimum) commutates a brushless motor in response to the state of user controlled buttons and dials. However, this computational mapping between user action and tool action is unchanging and unambiguous. In related work, researchers have examined systems that *record* how the tool is being used, *advise* users on how best to use the tool, *modulate* how a user is moving the tool, completely control the motion of the tool by *remapping* user input, or automatically fabricate *proxies* to be used without computation. Orthogonal to how these systems leverage computation is why. What is the purpose of the embedded computation? Does the system guide the user to a specific end result, or facilitate the process of using the tool?

2.3.1 Record

Recent research in HCI to create more fluid and interactive ways of working with digital fabrication seeks to retort established digital fabrication workflows. As was discussed, one of the earliest technologies for digital fabrication similarly sidestepped the need for complex workflows. *Record-playback*, it allowed a machinist to define a toolpath by simply manually machining the first part [84]. The way a machinist manually moved the axis of the tool could be recorded and replayed to create subsequent components. Over a half a century later, Willis et al. revisited this concept with *Cutter*, a foam cutting machine that records how the user controls the cutting mechanism [136]. Both of these systems utilized computation to augment the user in a simple way, by preventing them from needing to do manually create the same geometry more than once.

2.3.2 Advise

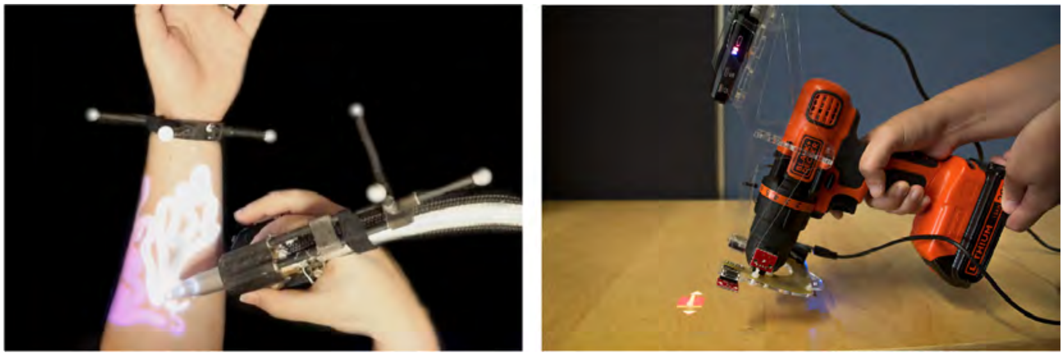


Figure 2.7: Exoskin [37] and Drill Sergeant [105]

Other interactive tools can advise the user during the fabrication process, but cannot actively fabricate for them. In *Sculpting by Numbers*, a combination of computer vision and visual feedback assisted novices during manual sculpting. As the user sculpts, the system scans the current state of the object, compares the scan to a desired digital model, and highlights (using a projector) where the digital and physical models differ [96]. In *Drill Sergeant*, one of the prototypes—a hand–drill—was outfitted with distance sensors and a pico-projector, both pointed at the workpiece. When the user drills holes with this modified hand–drill, information such as the current angle and depth of the drill bit could be visualized [105]. Feedback has also been explored for on-skin fabrication in *Exoskin*, a system that allows users to design and fabricate wearables [37]. In the fabrication workflow, the user can see the a projected visualization of where the design should be printed in relation to the body. Lastly, *Being the Machine* guided a user to 3D print by hand with materials of their choosing: coils of clay, balloons, or even pancake batter [25]. To indicate where the user should place material, the system used a laser pointer on an actuated pan-tilt mechanism.

In each of these projects, visual projection was leveraged towards diverging goals. Whereas *Sculpting by Numbers* and *Exoskin* visualized the proximity to a digital model, *Drill Sergeant* offered feedback that was not necessarily tied to a specific object to be fabricated. In *Being the Machine*, this modality was instead used critically and artistically to invert the delegative relationship between designers and the poster child of digital fabrication: 3D printers. Artifacts created with this system are shaped not only by the designer's intent, but by the tools, materials, and environments that fabrication occurs with.

2.3.3 Adjust

Figure 2.8: Enchanted Scissors [140], FreeD [144], Haptic Intelligentisa [45].

Going one step further, some systems actively modulate how a user works with a hand–held tool. The most well known examples are *FreeD* and the *Position Correcting Router*, both of which are hand–held tools where actuators can adjust the position of the tool's cutting head with respect to where the user is holding the tool [97, 144]. Combined with position sensing, achieved through magnetic and visual tracking respectively, these systems allowed users to precisely fabricate predefined geometries by hand. Prior to these two works, *Haptic Intelligentsia* explored a similar concept by attaching a hot glue extruder to a haptic controller. As a user moves the hot glue extruder in space, they can feel the contours of a virtual model rendered by the haptic controller [45]. Instead of modulating the position of the tool, *Augmented Airbrush* modulated when paint was allowed to be sprayed, again, based on the tracked position of the tool [110]. With *Comp*pass*, users could draw ellipses and squares with a drawing compass whose radius was actuated by a servo [78], In all of the above examples, the systems could adjust the actions of a tool based on a predefined digital geometry. Whereas some of these projects, such as *Comp*pass* and the *Position correcting router*, focused on allowing users to fluidly and accurately replicate the geometry, *FreeD* developed interactions that explicitly enabled users to modify and override the geometry during the process of hands-on fabrication. However, not all fabrication tools anchor the user to a predefined model: [140] created scissors which responds to conductive ink drawn by the user, *Protopiper* allowed users to build large sketch models using a hand-held tube extruder [1], and *dePend* developed a drawing system that can guide a user's pen with a magnet [139].

2.3.4 Remap

With respect to controlling position, hands-on control strategies discussed above limit the system to adjusting the action of the user, rather than controlling the action entirely. By decoupling the user from the tool, alternate strategies for mediating control and feedback can be explored. These strategies can continue to resemble hands-on control, but also can also become increasingly abstract.

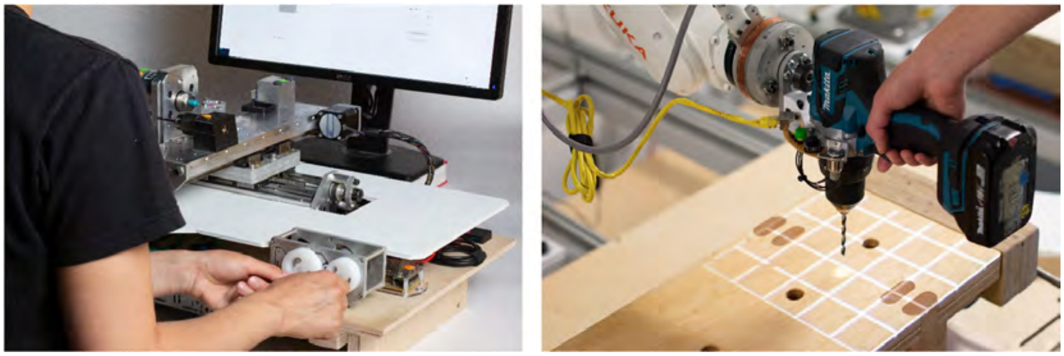


Figure 2.9: *Turn-by-Wire* [121] and *aDroid*

In *Turn-by-Wire*, a user controls a lathe through handwheels [121]. This configuration references how manual lathes are typically controlled: the user rotates a handwheel to translate a cutting tool. With *Turn-by-Wire* however, the coupling between handwheel rotation and tool translation is facilitated entirely through software. *aDroid* extends this concept for hand-held tools mounted on an industrial robot. To move the tool, the user simply grasps and moves the tool. Though the user appears to be moving the robot directly, in reality, the forces that the user exerts are continuously measured by a force sensor, and the system uses these measurements to calculate positions for the robot to move to. Both of these systems remapped user actions while leveraging haptic feedback. More importantly however, these interactions were utilized as a means for incorporating affordances of digital editing directly within physical fabrication tasks.

More abstract control strategies have also been explored. When a user draws on a touchscreen with LINC, the system conveys the user's strokes to a digital fabrication machine, which draws with a pen on paper [66]. Different strategies for when user actions are translated into machine actions were explicitly probed. A similar prototype from *Interactive Fabrication*, *Shaper*, allowed users to gesture on a transparent touchscreen to control where expanding foam should be extruded [136]. *Compositional 3D Printing* leveraged gestures as a means to modify, or "compose", the geometry of a 3D model as it is being printed [52]. Lastly, in one of the most abstract examples of continuous embodied control, *Speaker*, a user

Figure 2.10: LINC [66] and Shaper [136]

controls the geometry produced by a wire bender by speaking to it [136]. The system is not parsing voice commands, but interpreting the actual waveform of the user's spoken sounds into a geometry that is fabricated in real time.

Notably, none of the above systems prescribed a correct geometry for the user to fabricate. Instead, these systems demonstrated how computation can instead support the user during the process of fabrication.

2.3.5 Proxy

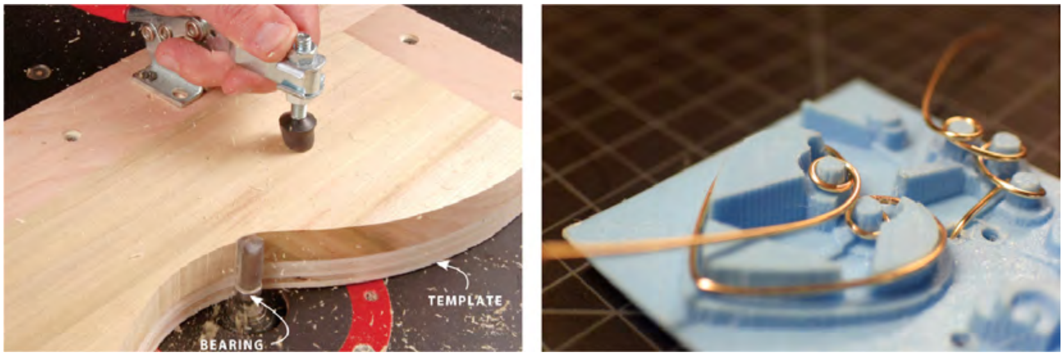


Figure 2.11: Left, physical router jig[1]. Right, ProxyPrint [123]

In hands-on fabrication, users frequently leverage jigs and fixtures. In essence, these are devices that allow a user to "copy" some type of geometry. For example, a router template allow a user to literally copy a contour from one piece of material to another. More simply, a drilling guide allows the user to "copy" the perpendicularity of the guide while drilling holes. Punch cards for Jacquard looms can be viewed through this lens as well—the loom "copies" the pattern to be woven from the pattern of holes on the cards. Researchers have explored

---

[1] Image retrieved from: `https://www.popularwoodworking.com/projects/template-routing-tips/`

how these intermediate tools can be computationally designed and fabricated. In these works, the user no longer interacts with computation directly, but through a physical proxy for the computation. For example, *ProxyPrint* explored jigs in the context of wire-wrap jewelry, and specifically examined the various roles that different types of static intermediate tools can play [123]; the term "proxy" is borrowed from this work. *JigFab* examined jigs to help build furniture with power tools [65]. In this system's workflow, the user first designs a piece of furniture using a CAD tool, and *JigFab* automatically generates designs for jigs which can be laser-cut and assembled. In *Hybrid Basketry*, a 3D printed formwork not only guided the weaving process, but also contributed to the aesthetics of the final artifact [143]. Recently, Albaugh et al. created a manual loom whose heddle mechanism—the portion of the loom which sets the weave pattern—can be computationally set by the loom as well as manually controlled by the user [2]. A range of interactions leveraging this unique arrangement were explored. For example, the system could record a user's manual adjustments of the weaving pattern, as well as support remote collaborative editing of computational patterns.

2.3.6 Beyond digital fabrication

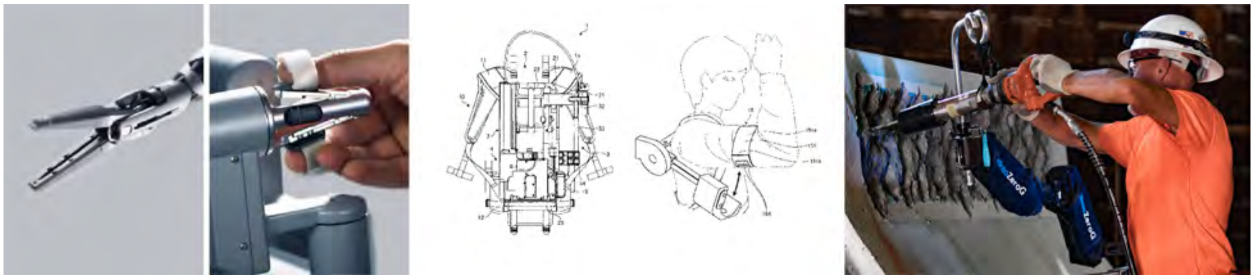


Figure 2.12: Left to right: DaVinci surgical robot[2][42], Makita patent for a battery powered exoskeleton [127], Ekso ZeroG spring loaded arm [30]

Outside of digital fabrication, most systems facilitate the *process* of using tools; the idea of a "correct" result in contexts such as surgery is more ambiguous. As HCI researchers continue to explore how hand tools can become "smarter" or "augmented" [146], the interactions and workflows proposed in these adjacent domains can help inform the design of new systems. Some technologies simply aim to give users extra strength. These systems can be as mundane as power steering in a car, or as experimental as exoskeleton "extenders" [51]. Commercial descendants [111, 30] of these extenders utilize passive spring loaded mechanisms to offset the weight of heavy power tools in construction and manufacturing. In film-making, similar mechanisms are used to smooth camera motions as well as offset weight [15]. In the field of medicine, numerous techniques have been proposed for mediating the use of surgical

---

[2] Image retrieved from: `https://www.davincisurgery.com/`

tools. Researchers have explored how tele-operated surgical systems can facilitate delicate motions by smoothing and magnifying a surgeon's hand gestures [42], "steady" the patients movements by synchronizing the robotic arms to the patient's heartbeat [79], or augment a surgeon's existing sense of touch through haptic feedback via sensory substitution [13]. Rather than controlling a tool from afar, some surgical systems allow the user and robot to simultaneously hold a tool for high precision tasks [117]. Though these works differ in context, they share a similar framing — greater strength, accuracy, or performance are desired, without sacrificing the unique capabilities afforded by direct hands-on control.

2.4 Conversational and computer numeric control

Conversational control shares a common objective as Continuous Embodied control: approximate the capabilities of digital fabrication and avoid the typical workflows. With Continuous Embodied control, users directly and continuously control the motion of the tool while being empowered by computation in some way. For Conversational control, the strategy is less about fabrication and more about design. Like a CNC machine, the geometry will be autonomously fabricated, but users are able to more succinctly specify the desired geometry to the system.

Though Conversational Control emerged in the late 1970s, similar approaches have been recently revisited by HCI researchers who have adapted it to new contexts, tools, and materials. In my own work, *MatchSticks*, this style of control was adapted for woodworking joinery. This work not only tailored the workflow tailored for joinery, but the physical machine itself [120]. In *Interactive Construction* and *Laser Origami*, users could interact directly with a laser cutter using laser pointers to specify where to cut, and what types of geometries to cut [75, 74]. Rather than having the user specify the entire design all at once to the laser cutter, this work allowed the user to iteratively create geometries.

Variations upon conventional CNC have also been explored. Work by Olwal et al. focused on feedback—during the autonomous operation of a CNC lathe, [82] enabled the operator to see the real-time forces subjected on the machine by the predefined toolpath.

2.5 Unpacking computer "aided" design

A full discussion of how computation can aid a user during the design process is outside the scope of this dissertation. However, it is still useful to contextualize fabrication tools with respect to the techniques that researchers have explored "upstream". Even though these techniques are more likely concerned with design rather than fabrication, they often share overlapping approaches and interactions as work in the fabrication tools themselves.

Many of the tools discussed in the previous sections supported the process of design implicitly because they do not require the user to have a fully determined object in mind before anything could be fabricated. In effect, objects could be designed as they are being fabricated. In these workflows, these fabrication tools could just as accurately be described as design tools. For simplicity of organization, this section will focus on tools which support a version of design where design and fabrication are less directly entwined.

Figure 2.13: Sketchpad [116]

In the early 1960s, Ivan Sutherland established the groundwork for Computer Aided Design with *Sketchpad*, the earliest example of an interactive graphics program. This research was conceived not only as a means to record drawings onto a computer, but to leverage sketching a medium for communication. In Sutherland's words, *Sketchpad* allowed "a man and a computer to converse rapidly through the medium of line drawings" [116]. Users where able to communicate the intent of their sketches beyond the literal marks that they made. This was made possible through geometric constraints. For example, if the user intends to communicate a square, they can start by roughly sketching a four sided polygon. After indicating that all corners should be perpendicular and that two adjacent sides should be the same lengths, the system can "solve" for the intended sketch, and computationally update what was actually sketched. If one of the sides of the square is lengthened, *Sketchpad* can solve for the constraints again to maintain the intended proportions of the square. In addition, *Sketchpad* could be used as input for additional computation. In Figure 2.13, Sutherland demonstrates how, after a user has sketched out a wire-frame bridge with appropriate constraints, the system can simulate the resultant forces in the bridge's members when a load has been applied.

The interactions and capabilities proposed in *Sketchpad* were prescient. Commercial CAD programs such as Fusion 360 or Solidworks all utilize constraints to allow the user to communicate the intended geometries, as well as to create a parametric model which can be updated after they has been drawn. These programs are also coupled with Finite Element Analysis (FEA), enabling designers to simulate stress, vibration, thermal, and other properties of a design before they are fabricated. Building upon these ideas, researchers have considered additional ways in which computers can “aid” the design process.

2.5.1 Optimizing the design

FEA simulations are an incredibly powerful way of predicting how a design will behave before it is fabricated. However, tuning the geometries of a CAD model based on the results from a simulation can consist of a repetitive cycle of updating the design, re-meshing the geometry, and re-running the simulation. The simulation itself can be computationally expensive, and slows this iterative cycle. Researchers have examined how to short circuit this process with systems that can simulate how the design will perform in real time as the geometry is being updated. In [128] for example, Umetani et al. created a system for designing lamellophones — planar structures that make a musical tone when struck. While the user designs the geometry, the system displays the associated vibration mode and plays the simulated musical tone. More recently, Schulz et al. created a system in which users can interactively view how modifying parameters of a CAD model will affect FEA results such as stress, deformation, or thermal distribution [107].

Alternatively, computation can automatically optimize particular properties of an existing geometry (such as moment of inertia [7], or buoyancy [131]) that would be difficult to tune manually through standard design tools. Designers are further separated from the task of defining geometries with techniques in topology optimization, which only need a user specify the requirements that the components must withstand — e.g. mechanical constraints, loads, etc. The resulting geometry is automatically determined by the algorithm [5]. Autodesk has recently incorporated this capability into Fusion 360, under the banner of Generative Design.

Much of this work is concerned with getting the design right, rather than getting the right design [122]. Simulation and optimization techniques are powerful tools for validation and fine tuning. However, these tools come into play only after a design is created. In other words, they assist the designer after the overall architecture of an assembly has been finalized, after material candidates have been selected, and after the manufacturing methods have been considered. By the time an engineer is simulating a design in FEA, they have already navigated a constellation of design decisions, and have committed the time required to create a geometry shaped by those decisions.

2.5.2 Constrain the domain

Designing objects on a computer can be challenging — it is an open-ended task conducted with an elaborate software tool. As a response to CAD/CAM pipelines that can fabricate anything, researchers have explored how systems could be instead be focused towards specific types geometries: custom joinery [60], structures from irregularly sized branches [142], interlocking [69] or volumetric [11] laser cut assemblies [69], or even other fabrication machines [33]. In many of these approaches, the design and manufacturing constraints specific to that domain are embedded into the tool, and low level details are abstracted away for the user. Work in industry has explored this concept as well. For example, *Vention.io* created a CAD tool for designing industrial automation mechanism with extruded aluminum profiles, and combined it with a service to deliver all of the required components.

2.5.3 Tangible input and feedback

Embodied interactions has been frequently suggested as an alternative to designing tangible objects through Graphical User Interfaces. For example, *HandScape* explored how dimensions could be transferred from the real to the virtual environments through an augmented measuring tape [63]. Building on those ideas, *Spata* utilized actuators calipers that can be used to specify dimensions in a CAD model, as well as physically render dimensions of a digital model [134]. Augmented with sensing, drawings on paper models [115] and physical construction kits [64, 4] could also be used as inputs to Computer Aided Design workflows.

Copying the geometry of real world objects as input for a CAD model was implemented in systems such as *CopyCad* [32] and *MixFab* [132]. Interactive mechatronic devices can be more easily designed by sculpting clay, using paper [104] and widgets [49] to annotate components (such as joysticks, potentiometers, etc.). After the sculpted geometry is scanned, they are computationally updated to accommodate the user annotated components.

Haptics has also been suggested for augmenting input for computer aided design [91] and manufacturing [72]. Other modes of input, such as in air gestures [137] and physical tools for mixed reality design [6] have also been explored.

Other works have explored the concurrent fabrication of a low fidelity model [77] in parallel with the user's creation of a CAD model [88, 89, 87]. Systems have also enabled bi-directional fabrication, where a physical model can be continuously edited by a machine [118], user, or both [133] after it is first created by a fabrication tool.

2.5.4 Tutorials

Software tutorials add another layer of assistance to Computer Aided Design. HCI researchers have broadly explored tutorials for complex software program. For CAD, researchers have explored tools that help instructors manage real time workshops [29], scaffold users from Minecraft [57], or understand workflows and suggest efficient ones to users in context [18].

Chapter 3

MatchSticks

With conventional digital fabrication workflows, a skilled designer can navigate CAD, CAM, and CNC software to make nearly any type of object. However, as we have discussed, this type of workflow can introduce a tremendous amount of friction into the design and fabrication process. Instead, how might users more directly interact with digital fabrication tools? Conversational Control is one alternative path that allows users to specify a geometry directly at the fabrication machine itself, rather than navigating CAD and CAM workflows in their entirety. To simplify the on–machine interactions even further, systems employing Conversational Control can constrain the range of geometries that can be fabricated. In this chapter, I explore this method of condensing conventional workflows for one domain in particular — woodworking joinery, the building block of wooden structures. In addition to focusing the software and interaction, I explore how a context specific fabrication tool can be designed to accommodate the unique geometric and fixturing needs of fabricating joinery.

The result of this approach is *MatchSticks* , a digital fabrication system designed to complement one of the first materials employed by humans — wood — and celebrate the fabrication practice of joinery. Combining a portable CNC machine, touchscreen user interface, and parametric joint library, *MatchSticks* enables makers of varying skill to rapidly explore and create artifacts from wood. This system distills the distributed workflow of CNC tools within the tool itself, operates on materials existing machines find difficult, produces assemblies much larger than its workspace, and supports the parallel creation of geometries. In this chapter, I describe the workflow and technical details of this system, present example artifacts, and report on results from a user study.

3.1 Introduction

Digital fabrication with wood often centers around using CNC routers to cut shapes out of planar materials such as plywood [97, 103, 65]. While this technique is capable of producing highly complex geometries, it does not appropriately address the geometries required by many woodworking tasks.

Figure 3.1: (Top Left) *MatchSticks* is a digital fabrication system for woodworking. (Bottom Row) Wooden objects larger than the size of the machine can be fabricated using a parametric joint library. (Top Right) We evaluate our system with users who followed an on-screen tutorial to create a parametric box.

Current Practice: Consider the chair in Figure 3.2 created using a traditional CNC router. To produce this chair from plywood, one must have access to a CNC router as big as the largest piece in the design (in this example, the size of the c-shaped leg sub-assemblies) or access to a much smaller position correcting router as presented in [97]. However, many of the cuts present in the design are straight lines that would be produced much more efficiently with other tools. The precision of the CNC router is only truly needed at the detailed joinery sites where pieces physically connect.

Figure 3.2: A chair design hosted by `Opendesk.cc`. Though the visual form suggests that the structure of the chair is created from lumber, it is in fact milled out of a large sheet of plywood.

While the constitutive components of the final model appear more like lumber, these geometries are created in a roundabout way by cutting down large, rectangular panels of plywood. Though this process appears inefficient, lumber (which can have slim aspect ratios resembling 1D stock material) can be difficult to work with using CNC tools, which are optimized for planar stock material. Recognizing this opportunity, and combined with our interest in hybrid approaches to making, we have created *MatchSticks*, a tool specifically for creating joints in lumber.

Creating furniture and the associated joinery is a hands-on and personal process that is a rich domain for the exploration of craft practices. Rather than focus on software that can create fully defined 3D models to be delegated to an omnipotent digital fabrication tool, we explore how a system of parametric joinery and a context specific CNC tool can give rise to interactions and affordances more similar to hand tools, augmenting the users' abilities while maintaining their autonomy.

3.1.1 Existing Fabrication Metaphors

Many digital fabrication tools lend themselves to prescribed methods of interaction. For example, the following properties are found in most traditional digital fabrication systems:

Siloed — The making process is broken into three distinct phases: users (1) design digital models using software, (2) convert that model to a machine parsable form (e.g. 'slicing', CAM programming, etc.), and (3) wait for the machine to produce the final object.

Geometry Agnostic — Digital fabrication tools provide equal form-giving consideration and focus across the entire work area without regard to the underlying intent of the design (e.g. sites of precise joinery vs. sites of straight line cuts).

Serial — A single machine is dedicated to operate across the entire stock material, neglecting opportunities for parallelism.

Limited Build Volume — Most digital fabrication tools can only make objects at the scale of the machine workspace.

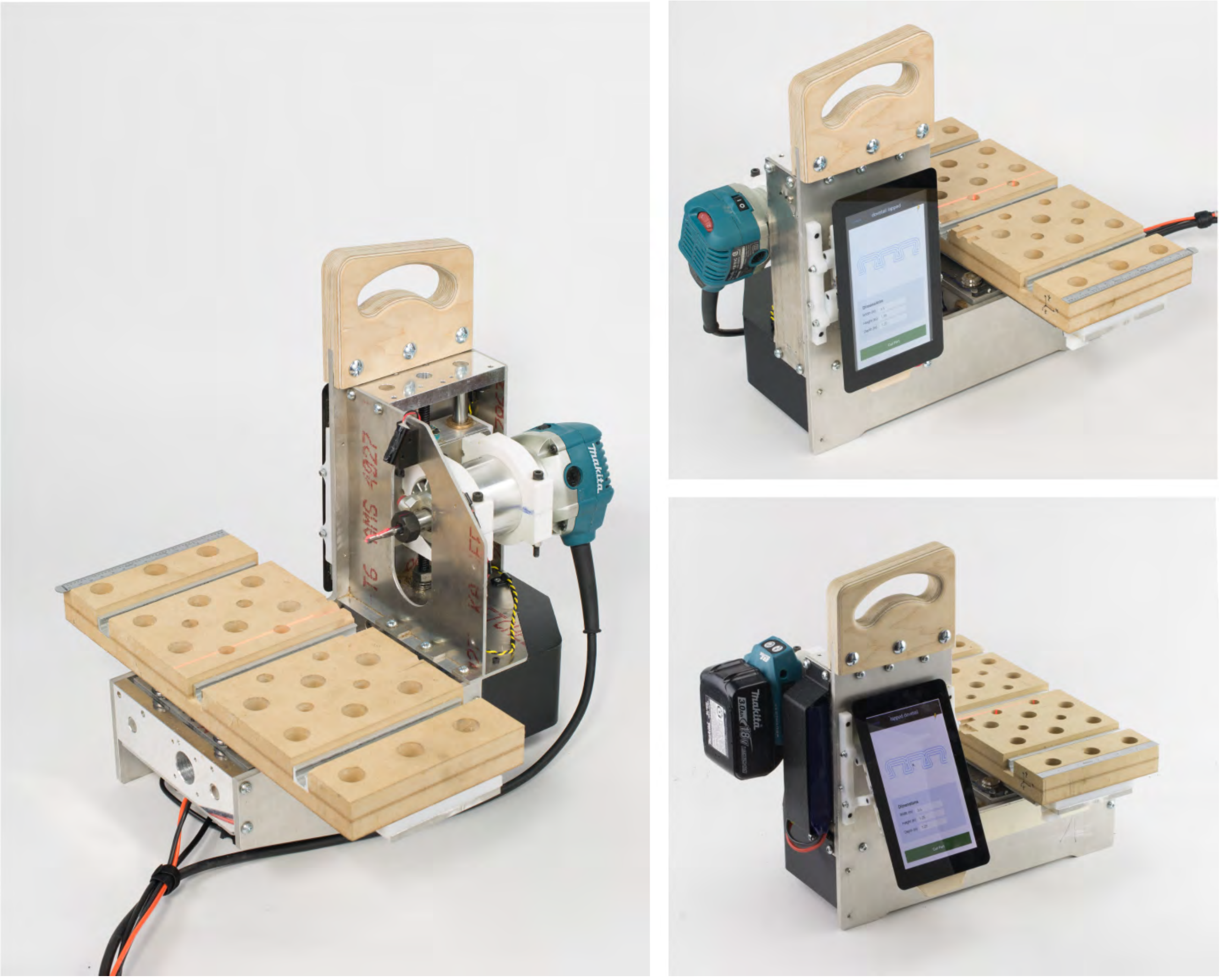


Figure 3.3: Our tool consists of a 3-axis CNC machine, trim router, touch screen for user interaction, line laser for quickly and accurately aligning cuts, and a modular build bed that accepts a variety of fixtures and alignment pins. During wireless operation, the motors run from a LiPo battery and a battery powered trim router is used.

3.1.2 *MatchSticks* Fabrication Metaphors

We adapt the traditional metaphors outlined in the previous subsection to enable a creative making process that is:

Interactive — We develop a more fluid and hybrid making [145] process that positions the tool and user as collaborators. Our design avoids the design-convert-wait workflow by positioning itself as a flexible smart workshop tool that is fully self-contained and can be independent of any external CAD or CAM software.

Geometry Conscious — For many structures (e.g. tables, benches, shelves, dwellings, etc.) the sites of highest fabrication complexity are at the joints. For our system, the capabilities of a CNC machine are invested exclusively into fabricating geometries that most require the accuracy and repeatability of Computer Numeric Control.

Parallel — We explore how a smaller set of specialized tools can be used more efficiently to outperform a single complex general purpose tool. Specifically, how a collection of networked, interactive, low cost, joinery specific smart-tools can be used in parallel to provide rapid fabrication capabilities and lead to new modes of human-machine making interactions.

Beyond Machine Scale — We leverage the localized nature of joinery to create tools that lend themselves towards making objects much larger than the tooling and workspace. Tools of this nature have the potential to enable individuals to rethink digital fabrication, expanding design and fabrication workflows beyond the workspace constraints embedded within current digital fabrication practice.

Envisioned Practice: As an example, consider a user designing and fabricating a similar chair to Figure 3.2, using a system that leverages these new fabrication metaphors.

Rather than designing an entire detailed 3D model of the chair on the computer, she sits down in the workshop with her materials and sketches a few ideas that capture the overall size and topology of the chair. Instead of delegating the entire fabrication process to the machine, she works interactively with the system and her materials to determine what joints would be best suited for the design.

She uses other woodworking tools such as table saws and miter saws in tandem, each used for the geometries that they are best suited to create. When she is ready to make a joint, the system guides her on how to fixture the pieces such that both sides of the joint will mate properly after they are cut.

When a mistake is made, she recovers gracefully, either re-cutting a single joint, or modifying the geometries of the chair to accommodate the happy little accident[1]. The mistake is known immediately after it is made, as components are iteratively created and assembled.

Because this system is much smaller and cheaper than other CNCs, the makerspace she is working in has many of these joinery machines. She is able to use more than one in parallel to create this chair. When a friend offers to help on her project, they work on the chair together.

---

[1] https://en.wikipedia.org/wiki/Bob_Ross

3.1.3 *MatchSticks*

We operationalize this philosophy in a new machine tool for joinery. Specifically, our tool comprises:

1. A novel machine tool capable of easily fixturing wood in various orientations to enable cuts on the edge and ends. Many of these geometries are substantially more difficult to create using existing tools due to undercuts and other geometric constraints (Figures 3.3, 3.5).
2. A touch screen display mounted on the machine as the primary interface for interacting with the tool (Figure 3.3).
3. A parametric joint library that can be scaled to lumber of various sizes. These files are stored as SVGs to allow the underlying representation of joints to be easily extendable by end users (Figure 3.7).
4. A toolpath generator to create machining toolpaths based on the SVG representation of joints.
5. A web application which handles the data storage, toolpath calculation and user interface, and allows the *MatchSticks* software to be agnostic to the underlying hardware.

3.2 Joinery

While it is possible to carve a bookshelf directly from a log, objects are often constructed from wood that has been sawn into boards to maximize material usage. As such, knowledge of how to join these lumber boards is essential to woodworking. Creating joinery requires both theoretical knowledge of the capabilities and aesthetics of joinery, as well as the tacit knowledge required in its construction. Many types of joinery exist, ranging from intricate double-blind dovetails to nailing boards together with a hammer. In this project, we turn our attention to joints that are stable with no additional hardware, which are valued for their strength and aesthetics. Example joints created by our system can be seen in Figure 3.7.

Many tools can be used to create joinery. Manual tools such as saws and chisels can be used to create nearly all joinery; their capabilities are limited only by the artisan's skill. Power tools are also used to create joinery. In addition to table saws, miter saws, and drill presses, many specialized jigs and tools exist. These joinery tools are often single purpose. To give a few examples, hollow chisel mortisers drill square holes, biscuit joiners cut shallow grooves used to align and reinforce simple joints, doweling jigs are used with a hand drill to make evenly spaced holes, and the Festool Domino[2] cuts slots for tenons. One of the most versatile power tools used to create joinery is a router: a rotary tool that removes material

---

[2] https://www.festoolusa.com/products/domino-joining-system

using spinning bits. In combination with templates, a router can produce joinery such as finger joints and dovetail joints.

Routers have also been integrated in flatbed CNC machines for woodworking use. However, these tools often have a strong preference for planar stock materials and one preferred orientation for fixturing. These constraints limit the geometry, aesthetics, and strength of joinery that can be fabricated.

In contrast, our tool is a CNC machine tailored for joinery. Compared to other methods for creating joinery, our tool (1) does not require a high level of manual skill to use; (2) does not require a large collection of power tools and templates for creating each type of joint, as joints are encoded in software, and output via a CNC stage; and (3) does not involve a large CNC machine that is ill-matched for joinery in lumber because of its size and preference for large 2D stock material.

3.3 Related work

Researchers have explored many approaches that enable users to create complex geometries with subtractive fabrication tools. We previously discussed two complementary approaches for interacting with fabrication tools: interfaces which mediate the continuous control of a tool, and those that allow users to more fluidly specify the desired outcome. This work is firmly within the latter category, the user is never literally guiding the tool by hand, but is nevertheless able to fluidly communicate their design intent to the fabrication system.

3.3.1 Machine design

In research and industry, novel manufacturing tools have been explored for a variety of applications. In order to examine how portability with precision can support new contexts for making, Peek et al. created *Popfab*, a multipurpose computer controlled tool that could fold into a briefcase [85]. In *Cardboard Machine Kit*, Peek et al. developed a toolkit which facilitated the process of creating custom computer controlled fabrication tools [86]. This toolkit included parametric hardware that could be composed into machines of different sizes and configurations, as well as networked electronics for controlling the motions of the tool. Both of these works are a part of the broader Machines that Make project, which has explored many desktop tools for digital fabrication, such as machines for printed circuit board (PCB) milling, wire electro-discharge machining (EDM), as well as tube carving [67]. These works were the inspiration for the custom fabrication tool in *MatchSticks*.

MatchSticks is a digital fabrication system tailored for wooden joinery — this goal influenced nearly all mechanical design decisions, including the orientation of the spindle, the layout of the gantry, the provisions for fixturing, and more. The layout and capabilities of our machine are most similar to the Pantorouter [130]. To cut joinery with the Pantorouter, the user guides a router through a pantograph mechanism that transfer geometries from a template to a workpiece. With *MatchSticks*, the templates and mechanical linkages are replaced with computation and actuators. In general, horizontal spindles are implemented in milling machine designs to maximize chip evacuation or to make cuts with fully supported tool [135].

Since this work was published at CHI 2018, other CNC machines for woodworking joinery have emerged, such as the Epur Oakbot (sized for timberframing) [31] and the Joyn machine from Studio Miltz [50]. Shaper Tools have also released an accessory, the Workstation, catered towards fixturing wood for joinery [109].

3.3.2 Computational design for joinery

Researchers have proposed a wide range of software techniques that automate or facilitate the process of designing furniture and wooden structures. For example, researchers have created systems that can deduce fabricable 3D models from ‘looks like’ 3D models [61], generate interlocking furniture designs automatically [34], and allow users to interactively design joinery with a focus on aesthetics [141] or manufacturability [60]. Researchers have also explored new aesthetics and materials: irregularly shaped branches where “upcycled” into structures with joinery in [142, 59], and wood as combined with 3D printed joinery to create new hybrid aesthetics in [68]. Within this constellation of related approaches, *MatchSticks* presents a CNC tool designed to fit within the existing ecosystem of woodworking tools. Like a miter saw, whose angles can be quickly adjusted for precise cuts, *MatchSticks* enables a similar fluidity and interaction for joinery.

3.4 System design

We detail the architecture of *MatchSticks* in the following subsections.

3.4.1 Mechanical design

The machine tool used in our *MatchSticks* system is a 3-axis CNC milling machine (Figure 3.4). Unlike most CNC mills, our router is held horizontally. This configuration, combined with the ability to easily fixture parts in a variety of orientations, allows us to make cuts that would be extremely difficult due to undercuts, fixturing, and constrained machining volume.

The y-axis (orange) carries the router up and down with a range of 3.5 inches ($\sim$90 mm) on linear round rails driven by a leadscrew. The z-axis (green) has a plunge range of 2.5 inches ($\sim$60 mm); it similarly travels on linear round rails driven by a leadscrew. The x-axis (blue) can travel 7 inches ($\sim$180mm), is coupled to the z-axis, and rides on V-groove tracks with rack and pinion drive. This configuration for the x-axis allows the length of the x-axis to be easily increased to machine continuous features along the edges of even longer boards.

The machine's kinematics are heavily over-constrained for maximum stiffness and minimum deflection under load. Components requiring high accuracy, such as the end blocks to

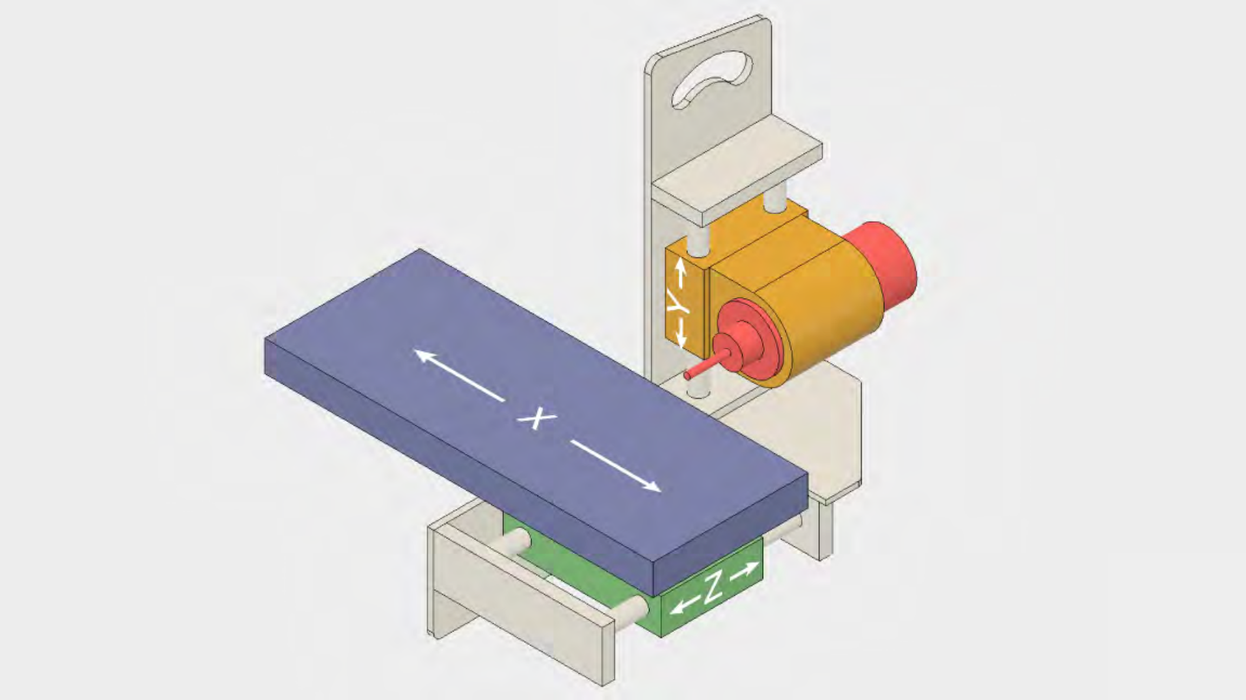


Figure 3.4: Mechanical layout of the *MatchSticks* gantry: y-axis (orange, 3.5 inch travel) carries the router (red); z-axis (green, 2.5 inches travel) plunges in-line with the router spindle; x-axis (blue, 7 inches travel) is coupled to the z-axis.

support the round rails, are hand machined from precision ground aluminum stock. The machine's reinforcing structure, whose tolerances do not need to be as accurate, is constructed from waterjet 1/4 and 1/8 inch aluminum plate. CNC machined MDF, HDPE, Delrin, and plywood are also used throughout the machine for the x-axis deck, spindle mount, bushing blocks, and handle and fixturing mechanisms. A standard off the shelf trim router is used as the cutting tool.

Mechanical design feature highlights

Size — While the bounding box of our machine is less than 1 cubic foot, the stock material that this machine can work with can be many times larger.

Portability — Our design intentionally tries to indicate affordances more akin to a hand power tool rather than a CNC machine. This is highlighted by the large wooden handle incorporated into the structure of the frame, allowing it to be easily moved throughout a workshop. The use of a battery powered trim router allows our device to function wirelessly.

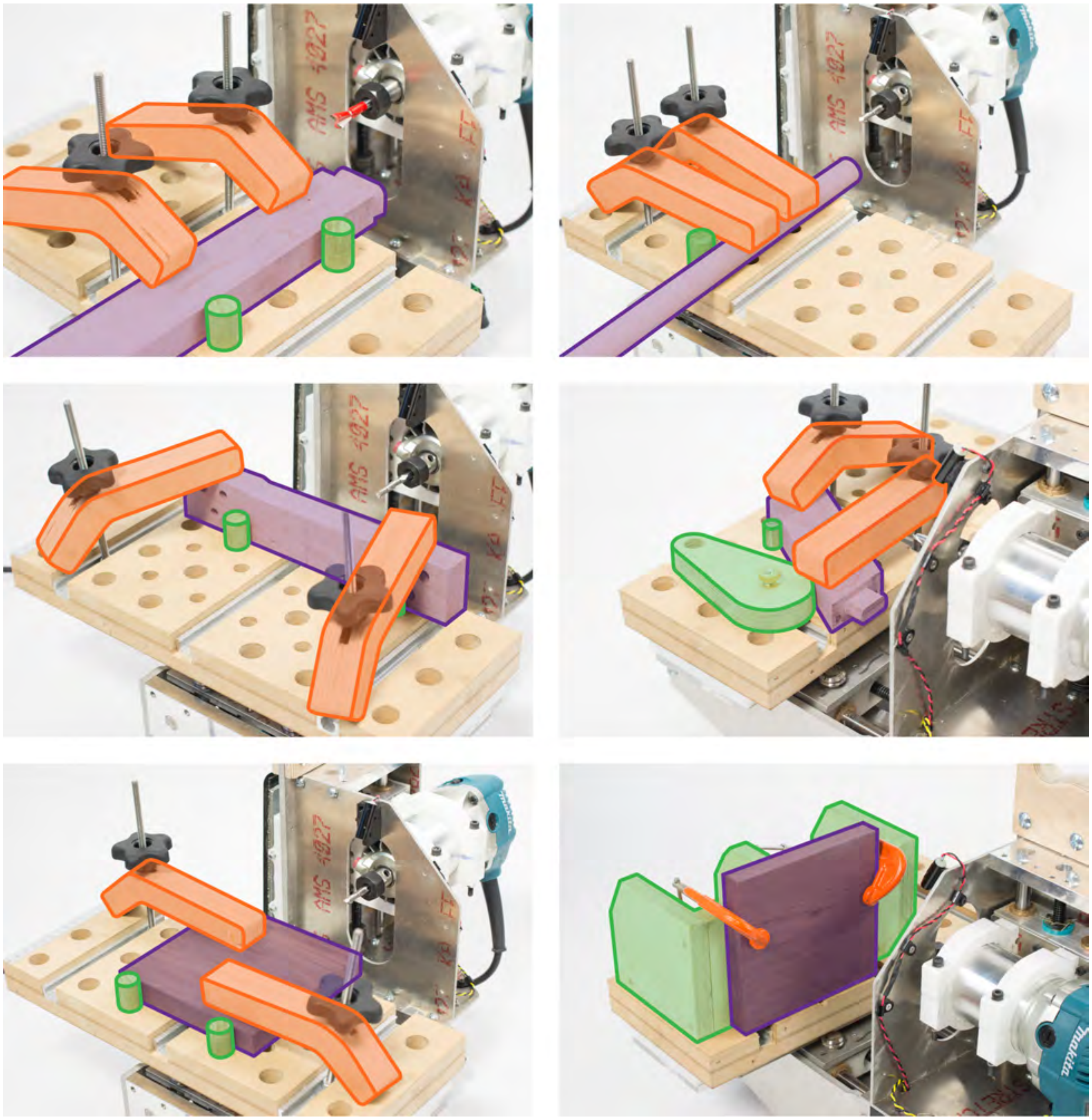


Figure 3.5: *MatchSticks* allows users to fixture components in a variety of orientations. This allows us to create geometries that would be substantially more difficult to create because of physical constraints such as undercuts. From top left: using our system, machining can occur at the ends of lumber, on round stock material, across edge profiles, at angles, in-line with the edge of a board, and on the surface of boards. This is made possible by modular components for fixturing (orange) and alignment (green). Workpiece shown in purple.

Fixturing — The deck of our x-axis supports a variety of ways to fixture wood. Our design incorporates T-tracks for clamps and reference holes for dowel pins used for alignment. Figure 3.5 summarizes multiple ways in which parts can be fixtured.

Zeroing — A line laser mounted to the frame visually indicates the current location of the spindle along the x-axis. This allows for fast and accurate visual alignment for cuts (in our experience, easily within +/- 0.5mm). While the joints themselves need to be very precise, the location of the joint along the edge of a piece of wood need not be as accurate, or can be compensated for by tracking reference faces (Figure 3.3).

Why create a custom CNC machine tool?

Operating on the ends and edges of boards is possible to do using a traditional flatbed CNC machine, but would require significant modifications to that machine. We chose to design our own for the following reasons:

Max Part Size — A board mounted on edge into a traditional flatbed CNC can only be as long as the distance between the floor and the router. Depending on the gantry configuration, a hole may need to be cut into the bed of the CNC machine. By placing the axis of the spindle horizontally, our machine can operate on arbitrarily long pieces.

Gravity — If the bed of a traditional CNC router is now vertical to operate on the edges of the board, fixturing pieces of wood held vertically can be difficult and error prone. Gravity is not on your side; fixturing parts precisely can be more difficult when the part wants to fall as soon as you let it go. With a horizontal spindle, boards are allowed to lie flat.

Poor Impedance Matching Between Machine Size and Feature Size — Flatbed CNC machines are often large in order to accommodate the entire piece of stock material, for example, a standard sized sheet of plywood. Joints themselves however, are significantly smaller. Our machine is designed to exploit this fact.

Unique Affordances — Most importantly, by designing our custom CNC machine, we have complete control over its aesthetics, affordances, and capabilities. It can be designed to be portable, extendable, modular, or even battery powered.

3.4.2 Electronics

When the device is powered through the wall, we use a 24V power supply to power the motors, regulated to 5V for the control electronics. In battery powered operation, we use a 3300mAh 4 cell LiPo battery and a similar battery powered trim router. The gantry is driven by three stepper motors. An Arduino running Grbl, an open source motion control software for CNC machines,[3] handles motor control. A Raspberry Pi 3 communicates with the Arduino, streaming G-code and reading the state of the machine. All major computation such as storing joint information, generating G-code, controlling the machine, etc., is handled through a web application rendered by the Raspberry Pi and displayed on a 7 inch touchscreen mounted to the machine's frame.

---

[3] Grbl: `https://github.com/gnea/grbl/`

3.4.3 User interface

Unlike hand tools that have no interactive interface, or digital fabrication tools that depend on external computers, *MatchSticks* is designed to afford direct, immediate, and creative workflows as directed by the user through handed interaction with the material and the machine.

The touch screen user interface (UI) serves to localize interaction with the tool. Both the design and fabrication of a part are directly at hand, rather than diffused to multiple workstations, software packages, or moments in time.

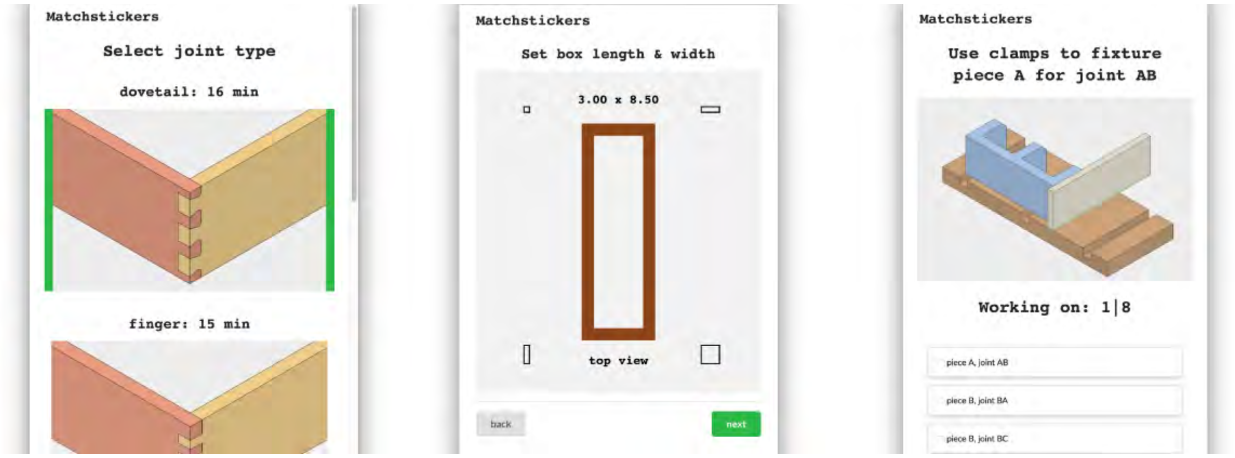


Figure 3.6: User interface for selecting the joint type and inputting the size of the stock.

The UI was built using the Sinatra DSL[4], which facilitates distributed services like downloading new joint types. The system is designed with modularity in mind – in future work, we envision many of these machines working together in a distributed manner within a workshop. The same UI can support machines with different routing characteristics and limitations. Further, each machine has access to the global state of the project: which joints have been made, and which still need to be made. The UI can guide the user step-by-step through various joinery workflows.

The vector image of the contour and toolpath are rendered using Paper.js[5]. To generate the toolpath, the outer contour of the cut is incrementally offset inwards using the Javascript Clipper[6] library. These offset contours are discretized and used to generate the toolpath G-code. Once the user finalizes their design, the generated toolpath file is streamed to the Arduino over a local serial connection.

---

[4] Sinatra: `http://www.sinatrarb.com/`

[5] Paperjs, vector graphics scripting framework: `http://paperjs.org/`

[6] Clipper: `https://sourceforge.net/p/jsclipper/wiki/Home`

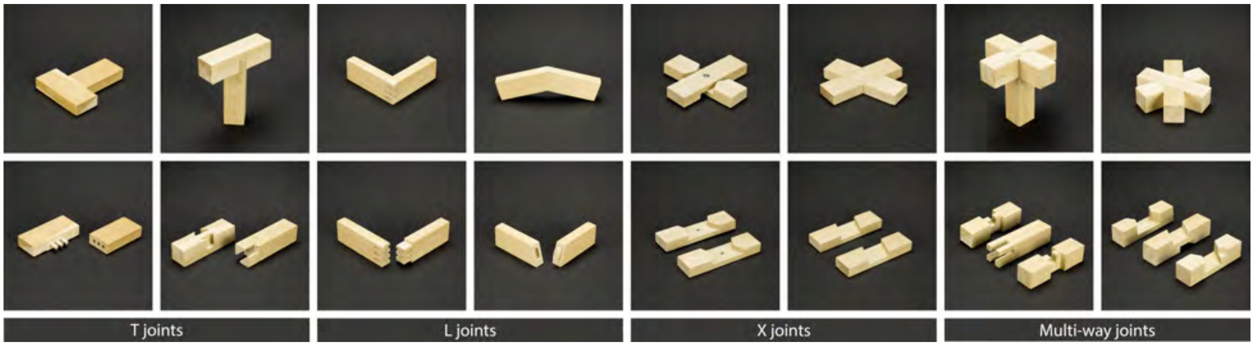


Figure 3.7: We categorize the joints our machine can create based on the orientation in which they come together. Top row shows assembled joints, bottom row shows disassembled joints. From left to right: dowel pin joint, framing joint, finger joint, angled mortise and tenon, pivoting lap joint, angled lap joint, three way interlocking tenon joint, and three way lapped joint. All example joints are created from poplar.

3.4.4 Joint primitive library

We categorize the joints our system can create based on where they occur on the pieces to be joined: end to end (L joint), end to middle (T joint), and middle to middle (X joint). Multi-way joints are categorized separately. In Figure 3.7, we show example joints in these four categories. We present these eight joints as examples; they are not an exhaustive list of the joints that can be created by our system.

The categories in this taxonomy address the high level design decision of ”What are the ways I can put these two wooden pieces together in this position?” This question is encountered frequently when creating wooden structures. The taxonomy introduces some redundancy in the style of the joints between categories, for example, a lapped joint (Figure 3.7, column 4) can be easily modified to become an L or T joint; this redundancy allows joints to be found when asking the question above.

While our CNC machine cannot capture all of the joints that can be made by a skilled artisan with all their tools and skills, our machine is capable of creating many commonly used joints, and creates opportunities to fabricate joints which are substantially more difficult otherwise.

3.5 Workflow

3.5.1 Open-ended design workflow

Recalling the hypothetical described in the introduction, we concretize the example in the context of *MatchSticks* .

The user begins with a high level design sketch. Using the atomic operations described below, she cuts a first joint and its mate. After inspecting the result and evaluating how it fits with her larger design vision, she iteratively cuts additional joints and assembles the final piece.

In this workflow, minimal guidance is given to the user beyond the available parametric joint library. The joints are arranged within the X, L, and T hierarchy for easy navigation when only the high level design intent is known. Other hierarchical groupings of joints that allow for rapid traversal of the design space can also be explored, such as grouping by the type object the user is trying to create.

This workflow is in some ways similar to the use of a miter saw, in which the user sets up a fixturing and chooses a machine setting, then performs an atomic operation on the material. With this machine, the user positions and clamps the wood in place, but rather than choose what miter angle to cut the workpiece at, the user chooses the type of joint that they would like to place on the piece of wood. The machine then cuts out that joint when it is commanded by the user. Like traditional woodworking tools, the design effort of creating the high level topology of the object lies solely with the user.

The user of this workflow should have some familiarity in designing objects, and the high level design can simply be sketched out by hand, or created using a lightweight CAD program. Because the complexity of the joinery has been abstracted into our system, we speculate that the inflection point for when a user would design by simply sketching out a model on paper versus using a CAD software will shift such that only very complex designs will need CAD modeling. Experienced users are also able to upload custom joints.

3.5.2 Cutting a Joint

Like all CNC machines, our tool can repeatably home into a known location. Therefore, the system is always aware of the height of the bed, as well as the location of the various holes for alignment pins. A zero plate is available for use when manual zeroing is required.

To cut a joint after it has been selected by the user:

1. The user inputs the dimensions of the piece of wood. The joint geometry is modified for this piece of wood.
2. Our system graphically indicates how the wood should be fixtured on the bed, so that the machine knows the piece's location.
3. The user confirms that they have fixtured the wood appropriately, and commands the system to make the cut.
4. The geometry is milled out.
5. The joint that will mate with the just-created joint is stored by the system for future operations.

3.5.3 Tutorials

The graphical user interface allows us to explore tutorials for woodworking projects. In addition to aiding the user through the fabrication of a single joint pair as described above, our system provides contextual information on how long the stock material should be, how and where to fixture the workpiece, zeroing the work if necessary, and in what order the cuts should be made. We have created one such tutorial for fabricating a parametric box whose joint types and geometry are customizable. This tutorial was generated by the authors, but in the future could be inferred from 3D models of objects. By simplifying one of the most difficult elements of woodworking, our system augments the existing design and making abilities of the user.

3.6 Example fabricated designs

Using our system, we created four example objects: two toolboxes, one side table, and one woodworking bench. These objects each display key capabilities of our system.

3.6.1 Toolbox 1

Figure 3.8: This poplar toolbox employs a variety of joints that would be difficult to create using current handheld and CNC tools.

The sides of this toolbox (Figure 3.8) come together at half-blind dovetail joints, a technique that requires considerable skill to produce using hand tools. With CNC routers that exist today, this joint is difficult to produce, and generally requires specialized fixtures. With our system however, this toolbox can be easily fabricated due to the ease with which lumber can be fixtured in various orientations. In addition, our software system removes the need to design and CAM the joint itself, allowing the user to more quickly move from design intent to physical artifact.

3.6.2 Toolbox 2

Figure 3.9: A variety of specialized router tools exist in woodworking. This example was created with a tapered router bit. The joints are highlighted by the alternating use of Peruvian Walnut and Birdseye Maple. While the boards used to construct this box have not been planed flat, the joints fit snugly because the contact areas of the joint have been *locally* planed (Left, locally planed area outlined).

In the design of Toolbox 2 (Figure 3.9), we demonstrate the ability of our system to machine hardwoods, use non-cylindrical router bits, work with non-rectilinear stock material, and create localized reference geometries. Many specialized router bits with irregular cut profiles are available for woodworking; for this toolbox, we used a flared router bit to produce the distinct sharp inside corners at the dovetails. Unlike dovetails cut using a dovetailing jig, we created dovetails with uneven spacing for a more interesting aesthetic. The handle is a maple dowel; the diameter at the ends has been reduced using a helical boring toolpath on our machine.

To create high quality joinery, stock material is typically jointed (made square) and planed (made parallel) before the joints themselves are cut. These operations are typically done with large machines that process the entire surface of the stock material. In contrast, the flexibility of our CNC machine allows us to *locally* surface the stock at the location of the joints such that the joint lines up perfectly, even if the underlying lumber has slight irregularities (Figure 3.9 left). Finishing steps, such as insetting the base of the toolbox or shaping the vertical components of the handle, are not fabricated using our tool, but with hand and power tools that are better suited for these details. This process integrates our tool into the broader ecosystem of the woodshop.

3.6.3 Side table

Figure 3.10: Poplar side table with acrylic top.

The side table (Figure 3.10) required all three types of joints classified in our taxonomy. The legs join to the bottom cross bracing with double tenons (T), and to the top cross bracing with dovetail joints (L). The cross braces use a simple lapped joint (X). All components of this table are longer than any dimension of our CNC tool.

Figure 3.11: This miniature workbench highlights complementary capabilities of traditional CNC routers and *MatchSticks*.

3.6.4 Mini woodworking bench

Our final example demonstrates a miniature Roubo[7] inspired workbench designed to raise workpieces above a standard worktable for a more comfortable working height (Figure 3.11). We use plywood for the tabletop, demonstrating the complementary capabilities of traditional CNC routers for 2D stock material, and our CNC router for 1D stock material. The legs supporting the top are machined by our system from 3x3 maple furniture squares.

---

[7] A style of workbench named after André Jacob Roubo (1739-1791), a master cabinetmaker and author of a highly influential treatise on woodworking.

3.7 Other machine configurations

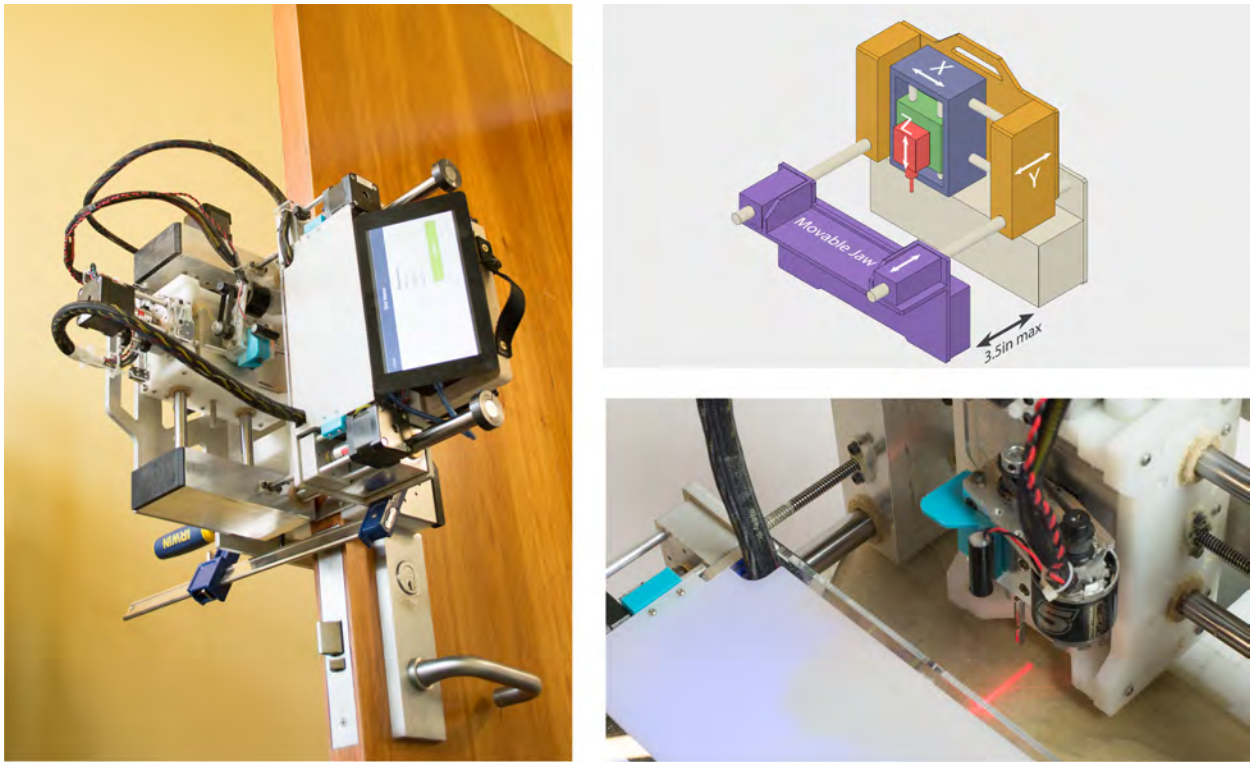


Figure 3.12: This variant of the *MatchSticks* device is designed to be clamped onto large pieces of wood that are difficult or impossible to move. Left, the device is set up for creating a mortise on a door for a deadbolt assembly in situ. Top right, diagram of kinematics, showing the movable jaw axis.

Building even larger structures may involve situations where moving the piece of wood is difficult or impossible either due to its size or its inclusion in an existing structure. In these scenarios, an alternate configuration of *MatchSticks* can be used (Figure 3.12), focused on mobility.

Rather than clamping workpieces into the CNC machine, the machine is designed to be clamped onto the workpiece. It is portable and battery powered to provide freedom of movement. In addition to the standard 3-axis milling head, there is an additional fourth axis integrated in-line with the y-axis which allows various pieces of wood to be clamped between the jaws of the machine. The linear rails for the vise axis are shared by the actuated y-axis of our machine, and the clamping force is provided by off-the-shelf clamps. As in the primary design, the motivation for this configuration is to access machining locations along the edges and ends of boards. While the kinematic configuration of this machine differs, there are many similarities in physical affordances, from the touchscreen that allows users to quickly access a library of joint designs, to the line laser and physical handle that allows the users to easily reposition the machine.

For this version, the majority of the structural elements are constructed from high-density polyethylene (HDPE) and aluminum plates and tubing that can be rapidly fabricated using a waterjet cutter and CNC milling machine. Bronze bushings and round linear shafts are used to create linear motion.

3.8 User study

Figure 3.13: Four user study participants completed a parametric box using *MatchSticks*.

We recruited 6 participants (four men, two women) to use *MatchSticks* in a 2 hour study in a local makerspace woodshop. All participants were registered users of the woodshop, with prior training on the basic tools and machines. All participants self-reported some hand tool experience, from intermediate to expert; digital fabrication experience ranged from none to expert.

During the study, users were introduced to *MatchSticks*, invited to fabricate a wooden box that they then took home, and interviewed about the experience. Four out of the six completed the entire user study and their box within the allotted 2 hours (Figure 3.13). Two completed an abbreviated, 1 hour study in which they cut joints but did not complete the box due to time constraints.

Participants followed the tutorial style described in *Workflow*, presented on the integrated touchscreen. Through the tutorial, each individual participant chose the dimensions of their box and the type of joints, cut stock material on a saw, milled their joints on *MatchSticks*, and assembled the final box by hand. We synthesize qualitative findings to evaluate our design and to inform future interactive tools.

3.8.1 Capability and accessibility

In prior projects, participants chose joining techniques based on speed, ease, or materials at hand. This tool opened up complex joinery as an option, challenging their perceptions of joinery as prohibitively difficult or intimidating. The woodworkers more experienced in joinery expressed renewed interest and confidence to attempt challenging projects:

> **P6** *[The tool] would lead me to do fancier projects. I'd be less afraid.*

> **P4** *When I was making 30 drawer boxes I was like [expletive] no I'm not going to do dovetails...but if it was an option, and I could just machine these...that's exciting.*

For some participants, this was their first experience with joinery. Not only did these novice users successfully complete the box project, but they now perceived joinery and CNC machining as techniques accessible to them.

> **P5** *If I was to make a box with only hand tools, I wouldn't have made dovetails or interlocking parts. [MatchSticks] makes more difficult things easier for me.*

> **P3** *If I were to make a box, I wouldn't want screws going into sections like this, so having a library to select joints from and entering my dimensions would be really convenient.*

> **P5** *I feel like I can do so many more things, like it gives me so many more options. [I had thought] CNC is...for when you're older and know more things and understand all the coding and stuff. This is eye opening because I realized CNC machines can help me even if I don't understand all of that.*

Quality was a key component in users' discussion of enjoyment and satisfaction with their woodworking projects. Both novice and expert users commented on the quality they achieved with *MatchSticks*:

> **P4** *I'm extremely impressed; I know how hard it is to get those tolerances, it's extremely hard.*

> **P3** *I like it, the edges are really straight, and it feels really sturdy, and won't turn into a parallelogram. It was a really nice fit, and all the edges are flush.*

3.8.2 Comparisons to other machines

After interacting with *MatchSticks*, participants characterized their experience in comparison to other woodworking tools they had used. They felt *MatchSticks* supported a more flexible, freeform, and natural workflow than a CNC machine while enabling greater capability than a traditional hand tool. Traditional CNC workflows are linear, slow, and inflexible, which participants found limiting.

> **P2** *I have to finish all the design on my laptop, then go there and do the work, and then try to make all the joints, see if it works or not. And then I'm done. I can't modify anything.*
>
> **P6** *You have to have it perfect before you hit the play button.*

But hand tool workflows are more flexible and interactive:

> **P1** *Hand tools are inherently more improvisational than a machine.*
>
> **P6** *With handtools and even with [MatchSticks], you can be halfway through and fudge something and get it there.*

After using *MatchSticks*, participants approached the CNC capabilities of the tool with an interactive mindset more akin to a hand tool. In particular, adjusting designs on the fly:

> **P2** *[With MatchSticks,] you can design during the fabrication process. If you make [a mistake], you can still make it part of the design.*
>
> **P6** *It reallly is an in-between. It's quick like a hand tool...and it's empowering [like] the CNC.*

3.8.3 Workflow and viscosity

Participants noted fundamental differences between a workflow with *MatchSticks*, and a workflow with a traditional CNC tool. *MatchSticks*' more flexible approach to design and fabrication inspired new perspectives on the relationship between tool, material, and design, radically different from a traditional CNC's design-convert-wait workflow.

> **P6** *[MatchSticks is a] really great conversational CNC type of device, where you have your drawings, you have your idea, you walk up to it and it kind of does a few calculations for you...I would not be able to walk up to a Tormach [CNC] and be like, 'Hey cut a joint for me.' It would be like, 'Give me some G-code, baby. I can't do nothing until you've done all the CADing and CAMing.'*

The material has a voice in this process as well:

> **P6** *You know what the biggest thing is probably, the use of scrap material in your project. Because when you're doing something with scrap...you end up changing things here and there as you move through it. So hand tools and maybe [MatchSticks] a little bit as well, you're working with what you have as you move along.*

The parametric library frees regular interaction with the tool from a long digital design process:

> **P3** *Doing my own CAD would be for my special use, if I had a special geometry and I had to design my own joint...but a library would cover what most people would want to build.*

3.8.4 Design of the machine

The affordances of the machine inspired new ideas for use cases, and increased comfort with the machine. *MatchSticks* can handle components far beyond the scale of the machine; one participant suggested its use in timberframing (the construction of buildings using wooden beams and joinery):

> **P4** *[the cut] would at the most be an 8 inch long by 2 inch wide…Those machines are huge that do that*[8]*. So the idea of bringing the machine to the stock and having the precision of a CNC is really exciting.*

Woodshops are inherently dangerous places. Some participants had avoided attempting dovetails or similar detailed work in the past because of how close their fingers would end up to moving blades. Users felt comfortable with *MatchSticks*, even in such a short time-frame, either commenting directly on its safety, or considering it equivalent to other tools.

> **P2** *It's so compact, it doesn't look dangerous. Some wood tools look kind of scary. It feels way more safe. If you see a huge bulky machine, even if it's doing a really simple thing, they kind of frighten the user.*

3.8.5 Satisfaction, agency, and craft

Finally, some participants reflected on the intangible aspects of working with wood, and the role of *MatchSticks*. One theme was direct interaction with the material:

> **P2** *It's different from calibrating a 3D printer. It feels different, I'm actively working with the material…You know how a miter saw is like a hand tool, but still 90 percent is clamping and making the dimension and then cutting it: even if the machine is doing all the hard work, I think I'm doing most of the work.*

---

[8] https://www.hundegger.de

A second theme was in the embodiment of craft. *MatchSticks* respects the knowledge and values of woodworking. The results maintain the aesthetics of traditional joinery, and the tool does not impart its own aesthetic onto the material.

> **P4** *The sensations and the muscle memory involved with using traditional tools is very satisfying...There's a soulfulness that's missing in digital fabrication... the nice thing [about this machine] is that the joints are pretty pleasing... I think that just doing that adds a level of satisfaction about the craft that gets really lost with those extra holes [in dogbones].*

3.8.6 Limitations

While many users felt empowered by their ability to create intricate joinery, some users with less or no prior digital fabrication experience felt detached from the actual making process:

> **P1** *The design is in my control, but the process is not. Which is good or bad, depending on who can do the process better - in this case it's definitely the machine. I do feel a little out of control of the fabrication process.*
>
> **P5** *When using a computer, it's convenient, but I don't understand what it's doing, I just know that if I type something in I know what's going to come out. With hand tools, I understand exactly what's going on.*

3.9 Discussion and future work

This tool is by no means an attempt to supplant traditional woodworking craftsmanship, or to replace other fabrication methods that work with wood. Rather, we are introducing a complementary process within the ecosystem of tools and techniques used for woodworking that is tailored for one of the most common operations one would do on wood—creating joinery with an adaptable palette of designs. By constraining a CNC machine to a very specific domain, and coupling it with a similarly focused software system, we have developed a workflow that can very quickly translate design intent to fabricated geometry. Reflecting on our fabrication of the example artifacts, this tool has greatly extended our own woodworking abilities. From the perspective of manual skill, this tool allows us to apply the accuracy and repeatability of Computer Numeric Control to fabricate geometries traditionally outside the scope of digital fabrication. Though we had to design many of these joints in order to populate the joint libraries, our user studies validated how a existing joint library can be used to dramatically reduce the viscosity of the design and fabrication of joinery.

The user studies also revealed a range of ways in which users imagine integrating a tool like *MatchSticks* into their workflows. The ways in which *MatchSticks* brings hand tool affordances to CNC capabilities encourages more interactive behaviors with the CNC tool, and more accessibility to joinery. Certain aspects of its CNC nature were off-putting:

exploring the roots of these reactions and how to address the perceptions of lack of control and understanding may be a fruitful path for future work in rethinking the role of CNC tools. In future work, we would also like to investigate how expert users within the domains of digital fabrication and/or traditional woodworking incorporate this tool within their practice.

In future work, additional *MatchSticks* devices can be created in order to explore interactions that can support a networked ecosystem of fabrication tools. Leveraging this infrastructure, methods for parallelizing fabrication, optimal planning, and collaborative workflows can be investigated. Imagine working with a group of friends to collaboratively design and fabricate a timberframed house, using a networked system of tools that are individually tailored for joinery, and collectively aware of the entire project state.

3.10 Conclusion

MatchSticks enables the fabrication of geometries and assemblies that are difficult to create using other tools, larger than the tool itself, and parallelizable during production. By eliminating the need to interact with a separate computer for the design of these joints, users can much more readily express their design intent to a digital fabrication tool. Using this system, novices can gain confidence and build proficiency with wood joinery as a material and technique, while experts are supported with an adaptive palette of joinery templates.

Despite these benefits, one limitation of *MatchSticks* (and digital fabrication tools more generally) is that users have little control over the actual process of fabrication. As corroborated by our user study participants, this way of controlling fabrication tools can be simultaneously empowering as well as disorienting. To address this opportunity, the remaining projects in this dissertation will transition from compressing existing workflows (Conversational Control) to exploring hands-on interactions augmented by computation (Continuous Embodied Control).

Chapter 4

Turn-by-Wire

In this chapter, I explore a set of concepts that invert many of the objectives that guided the design of *MatchSticks*. For example, *MatchSticks* allowed a user to more directly communicate a desired geometry to a fabrication tool, but how might a fabrication tool communicate back to the user? These tools are not omnipotent output devices that can create any design a user intends, but potentially a rich source for feedback about what geometries are possible to create, as well as the best methods for creating them. Second, whereas *MatchSticks* purposefully constrained the domain to allow users to draw from a manageable palette of geometries, how might a system support users in creating open ended geometries?

Intersecting both of these concerns is the question of how the machine should be controlled. Previously, we discussed one of the limitations of *MatchSticks*, the loss of agency that users encounter when they are no longer involved in the actual fabrication process. Delegating control to a fabrication tool can be desirable in certain circumstances, and troubling in others. In either case, the loss of control is often accepted to be a natural compromise of engaging with the capabilities of digital fabrication tools. In this work, I am interested in exploring how a user might continuously interact with a fabrication tool, while still being able to reap the benefits of digital fabrication.

To limit the scope of this work, I choose one tool as a case study to examine these ideas in detail — a lathe, a subtractive fabrication tool used for creating cylindrical objects. Users interact with *Turn-by-Wire* primarily through haptic controllers, which take the form of handwheels that are commonly found on manual machines. Through *Turn-by-Wire*'s handwheels, users can turn them to control the position of the tool; simultaneously, the lathe can communicate information back to the user through haptic feedback. As referenced by the title, these handwheels are coupled to the lathe through computation alone. The interactions enabled by this architecture are leveraged for three primary goals: perceiving state, communicating design intent, and scaffolding technique. I describe the technical implementation of this system and report on an evaluation with expert practitioners.

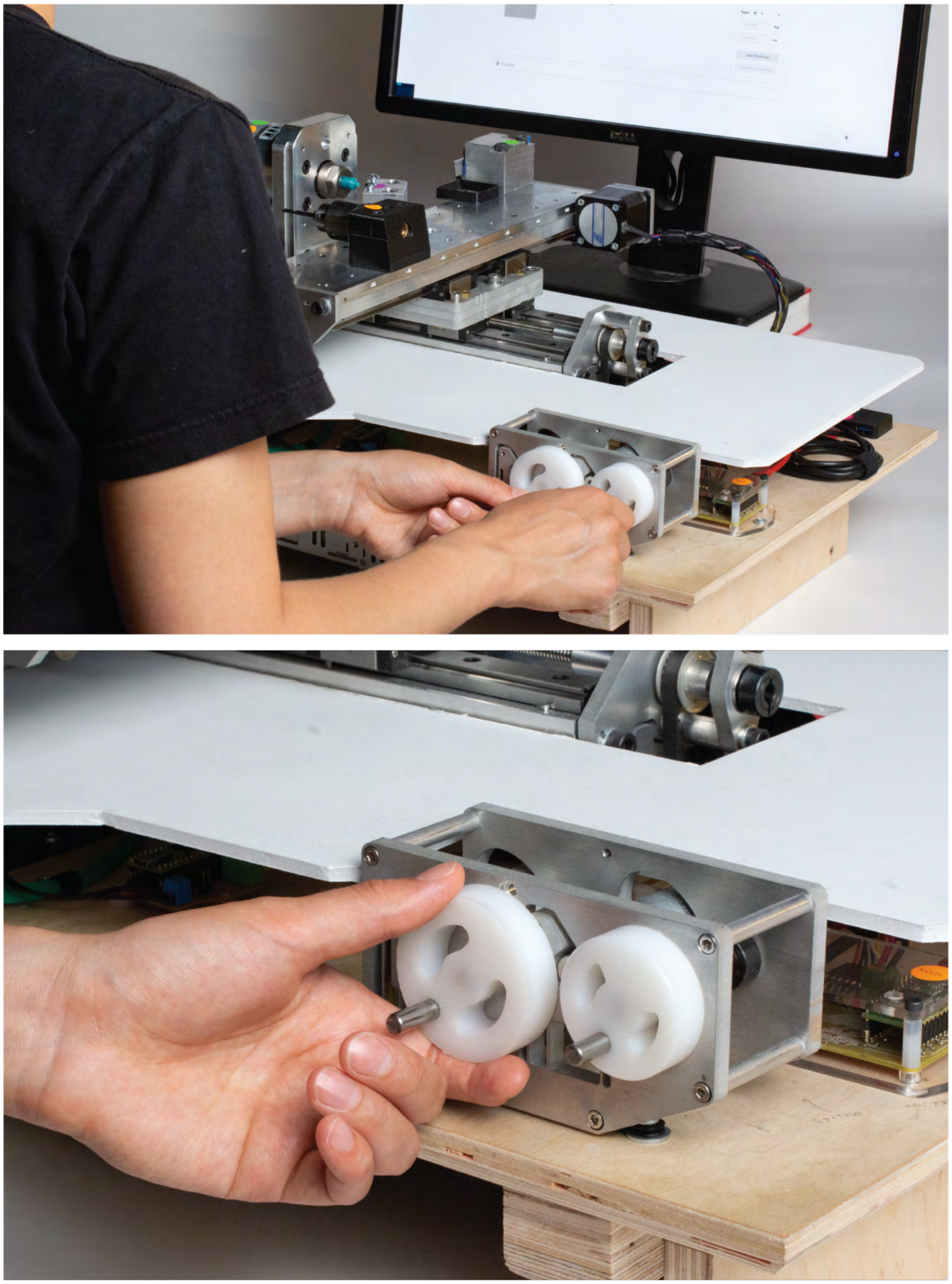


Figure 4.1: Top: the *Turn-by-Wire* system consists of a lathe, haptic feedback handwheels, and a graphical user interface. Bottom: detailed view of the handwheels used to directly and digitally control the lathe.

4.1 Introduction

Digital fabrication tools have many benefits: materials can be autonomously shaped, no manual dexterity needs to be developed by the user, extreme accuracy is achievable for highly complex geometries, and structured workflows are strictly enforced. While useful, these qualities of digital fabrication are not necessarily valued by all makers at all times. In some situations, other questions carry more weight: How might a skilled artisan incorporate their embodied knowledge within digital fabrication? How might we interact with fabrication tools differently if the maximum attainable precision and complexity is not always required? And how might these fabrication tools recognize that design constraints are often flexible and ambiguous, and be similarly flexible in their workflows?

Researchers in HCI and beyond have approached these questions from a variety of perspectives. The primary focus in this work is feedback — fabrication tools are not only output devices for creating physical artifacts, but sources of feedback that can and should inform the user about the state of their work and the capabilities of the system. Specifically, *Turn-by-Wire* focus on how a range of haptic feedback modalities can be leveraged to support users during manual fabrication. These computational interventions need not only glorify and recreate the sensations lost when transitioning from manual to digital fabrication, but leverage the digital medium to create new experiences altogether.

Broadly, we seek to create more *lucid* fabrication tools: those which respect our human capacity to skillfully and keenly respond during the process of fabrication, by allowing design intent to be fluidly expressed and feedback to be clearly understood. Though these ideas are generally applicable, I explore them in specific detail using a lathe.

4.1.1 The lathe as a case study

Lathes are fabrication tools in which the material being formed is continuously rotating, and tools move about the spinning material to add, cut, or deform it to the desired shape. The act of working with a lathe is called turning, which is referenced by the title. Lathes are historically as well as practically significant and often referred to as “mother tools” [40, 99]. They have endured because of their unique ability to quickly and accurately create cylindrical geometries (Figure 4.2).

Lathes are found in diverse contexts, exhibiting varying levels of handedness in how the tool is used. In the case of a ceramicist’s pottery wheel, hands are directly used *as* tools. One step removed from this directness is a wood lathe, where sharp tools are mounted on handles, giving the user greater leverage over the cutting forces and process. Going one step further is a manual lathe, where cutting tools are mounted to a two-axis gantry, and the tools are moved through the use of handwheels — large rotary knobs which transmit the rotational motion applied by the user to linear motion of the two axes through a mechanical transmission. In this third style of lathe, the physical transmission mechanism between the user and the gantry acts as rudimentary mediator between the actions of the user and the actions of the lathe. This decoupling between user action and machine action augments

Figure 4.2: Components made using our system, from right to left: Nesting dolls of three sizes, bowling pins, spinning top, chess rook, and Energy Dome. Ruler scale in millimeters.

the capabilities of the user by improving safety, productivity, and accuracy. It also reduces the directness of how the user works with the material, but the tactile sensations of working with the material are not entirely lost. Forces felt by the cutting tool are attenuated through the mechanical transmission. Fourth and lastly, CNC control takes this decoupling to an extreme, and all actions performed by the lathe must be specified through a series of software packages. This CNC workflow is not specific to lathes, but representative of the workflows for many other digital fabrication tools today.

This work is situated between the mechanical couplings of a manual lathe, and the disembodied control of a CNC lathe. *Turn-by-Wire* utilizes “drive-by-wire” handwheels to directly but digitally couple the user and the tool. A user directly moves the cutting tool through the use of the handwheels, but *how* the cutting tool moves is entirely software defined. The same is true for the forces that are felt through the handwheels; these handwheels act as the locus of interaction not only for expression, but also feedback. This “by-wire” infrastructure facilitates our primary contribution: the concept and implementation of a richer set of haptic feedback that mediates a user's interaction with a fabrication tool. These haptic interactions are not limited to literal representations of feeling a desired model or imitating the forces experienced by a tool. Instead, we leverage haptics to empower users to communicate intent, perceive state, and acquire technique during the fabrication process. For example, consider "snapping-to" a haptic guide, amplifying the forces felt by a delicate cutting tool, or learning a new technique through haptic guidance. These interactions and more are made possible by a system consisting of an electro-mechanical lathe, force feedback handwheels for physical interactions, and a graphical user interface for visual interactions (Figure 4.1).

4.2 Related work

With *Turn-by-Wire*, users directly control the motion of the lathe's tools through the handwheels, but the mapping between handwheel rotation to lathe motion is defined — and can be redefined — through software. A single turn of one handwheel can just as easily command a lathe tool to move linearly, or to trace a curve. Similarly with haptics, the lathe can communicate a simulated cutting force, or the position of a virtual guide. Multiple systems in HCI seek to allow users to rapidly specify design intent [75, 74, 120, 66, 136], or to give users additional feedback about the fabrication process [82, 45, 96]. *Turn-by-Wire* unites these complementary ideas.

Within computationally augmented hand tools, systems often guide users towards a pre-designed geometry [144, 96, 97, 45]. In contrast, *Turn-by-Wire* can enable users to create geometries that have yet to be designed, and utilizes computation to support the *process* of fabricating that geometry. With this approach, the boundary between design and fabrication becomes ambiguous. This is especially true when a user commands the system to record their actions: the act of machining a part with *Turn-by-Wire* simultaneously becomes the process for designing a model of that object, in-situ of the manufacturing constraints required to fabricate it.

4.2.1 Revisiting record–playback

Record-Playback, developed by General Electric in the 1950s, sought to combine handed familiarity with the potentials of numeric control by recording the actions of a machinist, and replaying those actions for subsequent parts [81] More recently, Haas Automation, a metal working CNC machine manufacturer, developed an electronic handwheel accessory for their CNC lathes [43]. Unfortunately, it offers little beyond imitating the experience of manual machining. The most important distinction of this style of interaction is that rather than attempt to tighten the interaction loop of programming and machining, the embodied actions of manually machining *is* the programming. Further, computation does not replace the embodied knowledge of a machinist working with the tool, but augments it in a simple way, through automatic replication. *Turn-by-Wire* builds upon this seminal work to further consider how computation can augment and extend a user's capabilities during the process of hands-on fabrication.

4.2.2 Haptics for digital fabrication

Tangible Holography, developed in the late 90s, was an early system that combined haptics with digital fabrication. Using a haptic stylus, users could cut a geometry on a virtual lathe while feeling a simulated reaction force [91]. More recently, *Haptic Intelligentsia* combined the haptic stylus with a physical tool — a hot glue extruder — rather than a virtual one [45]. Whereas in [91], haptics communicated information about how a material reacts to a tool, [45] used haptics to communicate the shape of a predefined digital geometry. Related to the idea of haptic feedback relative to a predefined geometry is *Enchanted Scissors*, which can prevent users from cutting into areas marked by conductive ink [140]. In a way, these scissors are reacting to a "model" that has been defined, but a user can modify that "model" directly and rapidly with a conductive ink pen. Outside of digital fabrication, haptics has been explored as a method to communicate "feel" during tele-operated robotic surgery [95]. In robotics more generally, haptics were used to render virtual fixtures that aided in a tele-operated peg-in-hole insertion task [100].

Researchers have previously utilized haptics within digital fabrication, but these approaches focused on either recreating the experience of using a physical tool, or feeling the contours of a predefined model. By leveraging the "by-wire" control fully, *Turn-by-Wire* demonstrates haptic interactions that extend beyond the literal and model-centric interpretations of what a user can feel through a computationally mediated manual tool.

Rotary haptic controllers map naturally to the mechanical handwheels that are normally used to control manual lathes. However, if we consider how haptics could intersect with other types of digital fabrication tools that have less obvious handles for control and feedback, it is helpful to assess a broader range of haptic feedback techniques and modalities. For example, researchers have explored variable stiffness springs for hand-held devices [41], texture [12] and object [54] rendering for Virtual Reality (VR), and lateral skin stretch [92] for semi-autonomous driving. Additionally, un-grounded force feedback has been implemented through flywheels [8] and asymmetrically accelerated voice coils [20]. This sparse sampling highlights the breadth of modalities that could potentially be leveraged.

4.3 Interacting with *Turn-by-Wire*

Our system leverages the direct but modulated control of digital fabrication tools for three primary goals: (A) perceiving state, (B) scaffolding, and (C) communicating intent. We first describe a short example of how these features are used in the context of fabricating a component, then discuss these interactions in detail with respect to their primary goals.

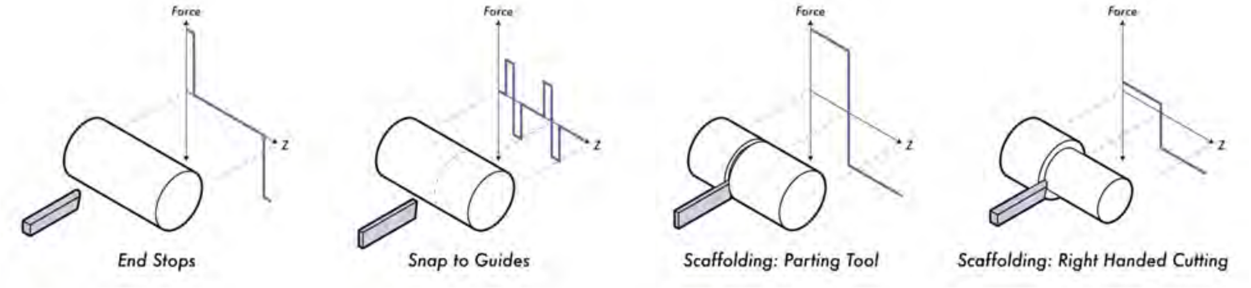


Figure 4.3: To illustrate the forces a user feel while using our system, we qualitatively plot the force with respect to the z position of the tool. The cutting tool is shaded in blue, and the cylinders represent the stock material to be cut. From left to right: Feeling the end stops, snapping to guides, haptic scaffolding while using a parting tool (tool cannot move in the z-axis after it has plunged into the material), and haptic scaffolding while cutting with a right handed tool (tool can only move right to left).

4.3.1 Using *Turn-by-Wire* to cut a simple shape

Our user needs to make a cylindrical component for the robot she is building. To do so, she selects a piece of stock material of the appropriate length and diameter, and loads it into the lathe. Through the user interface, she inputs the dimensions of the stock material, establishing the initial correspondence between virtual and physical material. To create the outer diameter of the geometry, the user wants to use the right-handed cutting tool, a common tool used for general-purpose cutting. She selects the tool from the GUI, and the position control of the lathe reciprocates from her to the machine as the system moves automatically to complete a tool change; handwheel controls are locked out while the machine is moving autonomously. When the tool change is complete, the user regains control of the lathe through the handwheels.

Using the 2D visualizer in the GUI, she sets up a guide-line to her desired dimension before starting the cut. Outside of the bounds of stock material, she turns the radial axis handwheel until she feels the handwheel click into the guide location that she has set on the GUI. She nudges the handwheel from side to side and feels a gentle centering force. Satisfied that the diameter is precisely set, she turns on the main motor to begin cutting.

She begins her cut by moving the tool along the z-axis of the spindle. As soon as the cut begins, she feels the resistance in the handwheel. Through her experience working with the machine, she knows that this amount of force corresponds to a safe cutting speed. As the physical material is being cut, the virtual material's geometry is simultaneously updated. Toward the end of this operation, she notices that she can no longer advance the z-axis any further – there is a virtual wall blocking her way. She turns off the main motor to examine the setup more closely. She sighs a breath of relief; it has been a while since she has used her lathe, and she had almost moved the z-axis into the frame of the lathe.

A few more operations later, our user finishes the component. Knowing that it would be good to have a backup of this part, she sets up the lathe to automatically duplicate the component, using the toolpath she just generated through directly interacting with the machine.

4.3.2 A - Perceiving state

The force of a cut is felt through the handwheel by electronically braking the motors. In addition, these forces can be amplified or attenuated depending on the task at hand. For example, a user might might choose to amplify the force on a delicate cutting tool to give themselves a more nuanced feel of how the cut is progressing. Physical attachments for manual machines are often created to accomplish this same task for small diameter drills. Rather than having the first indication of a broken or failing part be the failed part itself, a user of this system immediately *feels* that something might be awry.

4.3.3 B - Scaffolding

While haptics may guide the user to better understand the state of the machine, we also consider how it can be leveraged to scaffold novices into a new domain, or for experts to learn new techniques. We create two haptic interactions to conservatively bound the user within safe operating conditions: limit stops and technique scaffolding. Users can toggle these features through the GUI to adapt the tool to their skill level. While these interactions restrain the allowable input of the user, they also seek to foster an environment where open ended exploration is encouraged and de-risked.

Limit stops

The position of the lathe with respect to the joint limits is known at all times. When the user begins to approach the travel limits, a virtual spring renders a repelling force, indicating to the user that something limiting the motion of the lathe is close by. When this feature is enabled, the chance of the tool crashing due to an error in setup is dramatically reduced.

Technique scaffolding

Lathes employ various types of tools for different operations. Similarly varied are the techniques for using those tools: preferred direction of cut, where or how to initiate or exit a cut, etc. While some of these techniques are a matter of whether or not the tool will cut at all, others are relevant for producing tighter tolerances or better surface finishes, and others yet are matters of personal style. When technique scaffolding is enabled, we restrict the user's input to the lathe through haptic feedback. One example of a strict rule check for technique is disallowing a user from moving a parting tool side to side after it has plunged into the material (Figure 4.3 center right). Doing so may damage or break the tool, depending on the material being cut. This restriction is indicated to the user by rendering a strong spring centering the z-axis handwheel after the cut has been initiated, which is released only when the user has backed the tool out of the material.

4.3.4 C - Communicating intent

Complementary to features oriented toward feedback, we create three interactions that allow design intent to be expressed more concisely. These interactions incorporate affordances typically associated with digital editing to augment the experience of physical fabrication.

Snap to guide

Virtual guides (detents) are added and visualized through the GUI (see Figure 4.7), and felt in the handwheels. Without visually interfering with the component being made, we afford a subtle mechanism for the user to track their current location in space.

Virtual tools

By using a digital rather than mechanical coupling between handwheel and lathe, we can adjust the "transmission ratio" between the handwheel's rotation and the lathe's motion in software. Rather than having the z-axis handwheel move the lathe only along the z-axis, this handwheel can be remapped to cut along a taper or radius, as opposed to coordinating two handwheels manually or by using a context-specific jig. Custom defined profiles can also be used. In the UI, this remapping is conceptualized as "virtual tools". The dimensionality of the task is effectively reduced to highlight the most salient aspects about the design.

DRY: don't repeat yourself

Copy and paste is an ubiquitous feature of digital tools, allowing users to cache and pattern motifs. We extend this concept to physical fabrication: once a object has been made by the user once, she can replay her actions autonomously, as in the scenario above. The design intent only needs to be communicated once, as all user inputs can be recorded.

4.4 Implementation

In this section, we describe the hardware, electronic, and software infrastructure that underlies our system. At a high level, the interactions are implemented using microcontroller-controlled hardware devices (lathe and handwheels), whose collective behavior is coordinated by a web-application that communicates to each hardware device through UART Serial. This architecture allows us to abstract the implementation of coordinated behaviors into a more high level programming language, while maintaining the timing reliability of microcontrollers

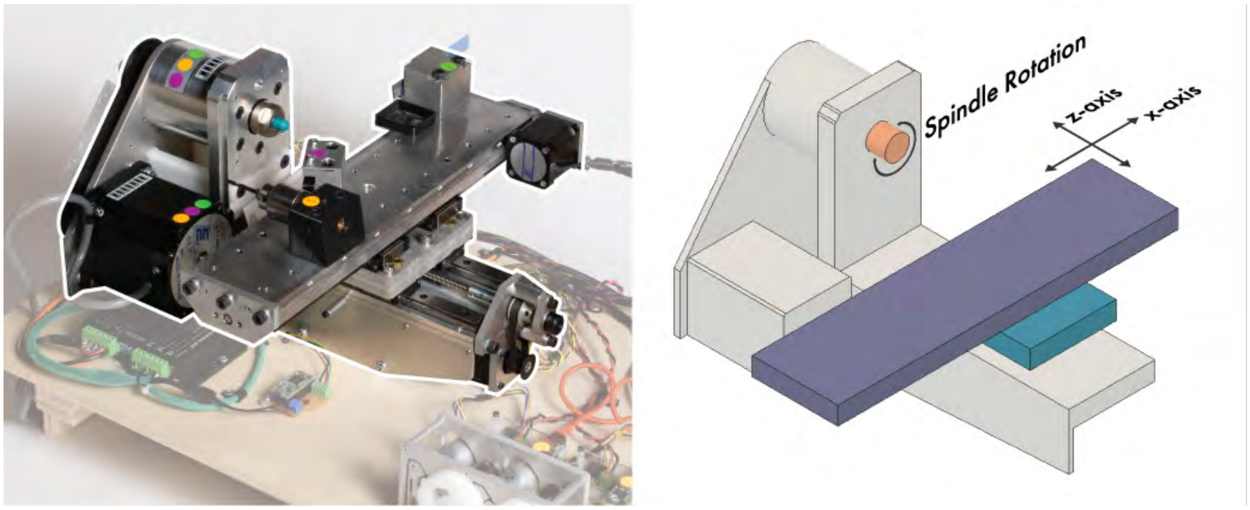


Figure 4.4: Left: The lathe is highlighted in a white outline to distinguish it from the background. Geometries to be cut are mounted to the spindle using a collet, which is currently holding a blue workpiece. Right: schematic view of the lathe. Rotating spindle in orange, x-axis in purple, z-axis in blue.

for control loop based interactions, such as haptic rendering. We highlight implementation details, discuss design motivations, and give a concrete example of how the system works together to operationalize the snap-to-guide functionality.

4.4.1 Lathe

We designed and fabricated a custom desktop lathe for our system. The lathe's structure is constructed primarily from 6061 Aluminum, using a combination of CNC machining, hand machining, and abrasive waterjet cutting. For ease of integration and assembly, we use linear profile rails to support the motion of the gantry. Stock material of up to 50mm ($\sim$ 2 inches) in diameter can be cut. A schematic view of the lathe can be seen in Figure 4.4.

A long x-axis (225mm $\sim$ 9 inches) allows us to simultaneously mount multiple cutting tools in a “gang tooling” configuration. Because the machine is CNC controlled, the user can easily index between these various tools, effectively using the x-axis of the gantry as an automatic tool changer. The following tools are currently mounted to our machine: a drill (used to create holes), a standard cutting tool (used to cut profiles on the outer diameter), and a parting tool (used for cutting grooves and removing the component from the stock) (Figure 4.4). These tools encompass many common operations possible on a lathe.

The spindle, which holds the component being fabricated, is driven by a NEMA 34 stepper motor as opposed to a brushed or brushless DC motor. This simplifies precise velocity and position control of the spindle. Though the maximum rotational speed is limited by the use of a stepper motor, respectable torques (approximately 4N-m at 100RPM) are achieved at lower speeds. The lathe's motors are controlled by a microcontroller running GRBL, an open source motion control firmware. G-code and other low level commands are streamed to this microcontroller over UART Serial.

4.4.2 Force feedback handwheels

We designed the visual layout of our two handwheels to reference the arrangement found on a traditional lathe: one larger handwheel for controlling the z-axis, and a smaller handwheel positioned to the side and offset above for controlling the x-axis (Figure 4.5). In a manual lathe, the handwheels are mechanically coupled to the motion of the lathe, and the user feels the forces required to move the lathe as these forces travel through the mechanical transmission. In contrast, our handwheels are drive-by-wire; they measure position and output force to the user, but there is no coupling between the handwheel and the lathe other than the ones we design and implement in software. This architecture is the basis for all of the interactions we have developed.

Mechanical design

Each handwheel subassembly consists of a handwheel mounted to a rotary shaft, a brushed DC motor with encoder, and a belt drive transmission coupling the DC motor to the handwheel shaft. A frame, consisting of two waterjet aluminum plates separated by spacers, house these components (Figure 4.5). The handwheel motors are mounted to flexural stages monolithically integrated within the frame; the thin flexural elements allows the motor to deflect slightly when the transmission is under load. In future work, we discuss how this deflection can be measured to directly calculate the force experienced by the user.

Electronic and firmware

Electronics and firmware is modularized by controlling each handwheel independently with a dedicated microcontroller. A custom printed circuit board interfaces with the encoder, motor, and other I/O. Each handwheel microcontroller exposes a UART Serial interface for commanding low level actions such as querying the current position, electronically braking the motor, or rendering a virtual spring at a desired position. More complex interactions are orchestrated by the web application.

Haptic control loop

The handwheels utilize two types of haptic feedback, electronic braking (passive) and virtual springs (active). For stability and ease of implementation, electronic braking is used to render continuous cutting forces. The cutting force rendered is proportional to the volume of material removed by the tool, which is calculated using the virtual model and tool. Active feedback is used for discrete features of interest such as virtual walls and detents, and is

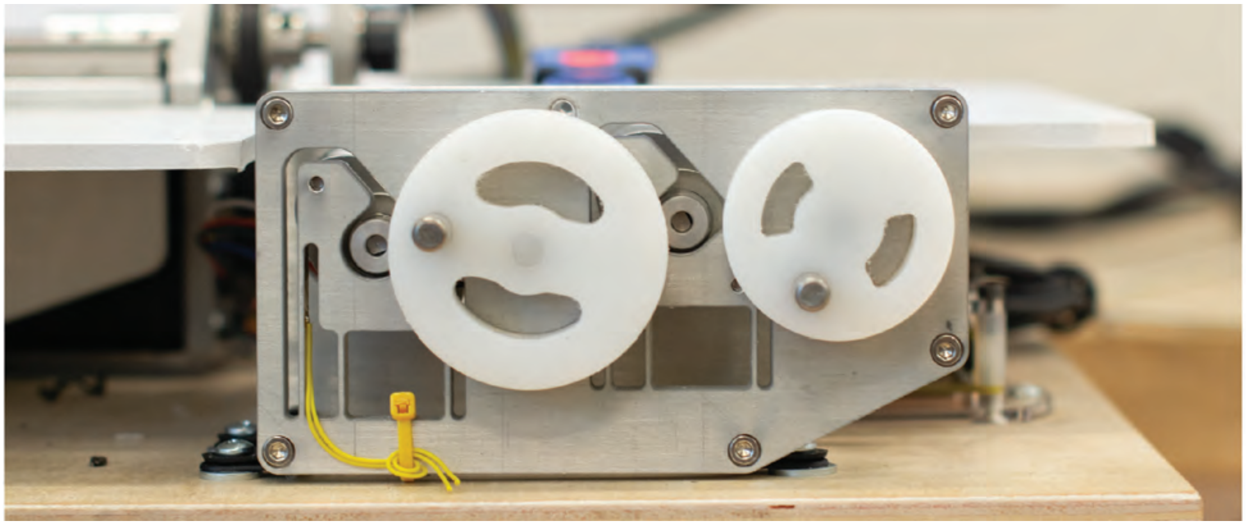


Figure 4.5: The arrangement of the x and z-axis handwheels references the placement on a traditional lathe.

implemented using an impedance (measure position, output force) type haptic controller [44]. The control loop only renders one type of haptic feedback at once, based on a hierarchy of the haptic interactions. For example, feeling a virtual guide takes precedence over feeling the force of the cut; when the user is cutting material but enters the range of a guide, the passive braking is disabled, and the user only feels the centering force of the guide. Though not implemented, an admittance type haptic controller (measure force, output position) could be used by measuring force through the flexural stage.

4.4.3 Orchestrating hardware through web applications

Using web applications to manage interactions between hardware devices affords two main benefits: the *Turn-by-Wire*system can be controlled by any device with a USB port and access to a web browser, and code orchestrating interactions between hardware devices is maintained in JavaScript rather than in microcontroller firmware. *Turn-by-Wire*uses two locally hosted web servers to coordinate interactions between the lathe and handwheels (Figure 4.6 center). The immediate interface to the hardware serial connections is a Serial to WebSocket bridge implemented using Tornado, a Python web framework. The UI is displayed on a computer monitor located adjacent to the lathe, and is served through a Sinatra DSL web server. We use Paper.js to render the top-down visualization of the lathe (Figure 4.7). The majority of the application logic is handled by front-end Javascript.

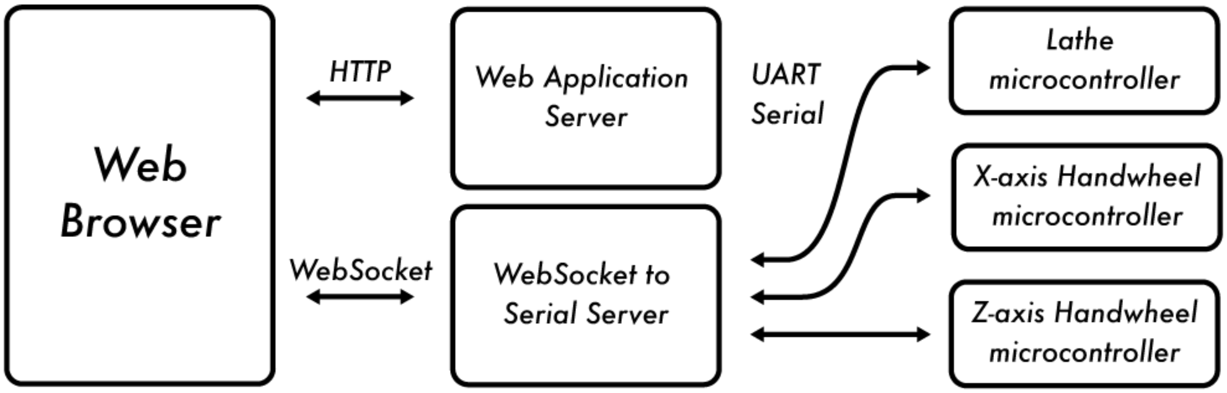


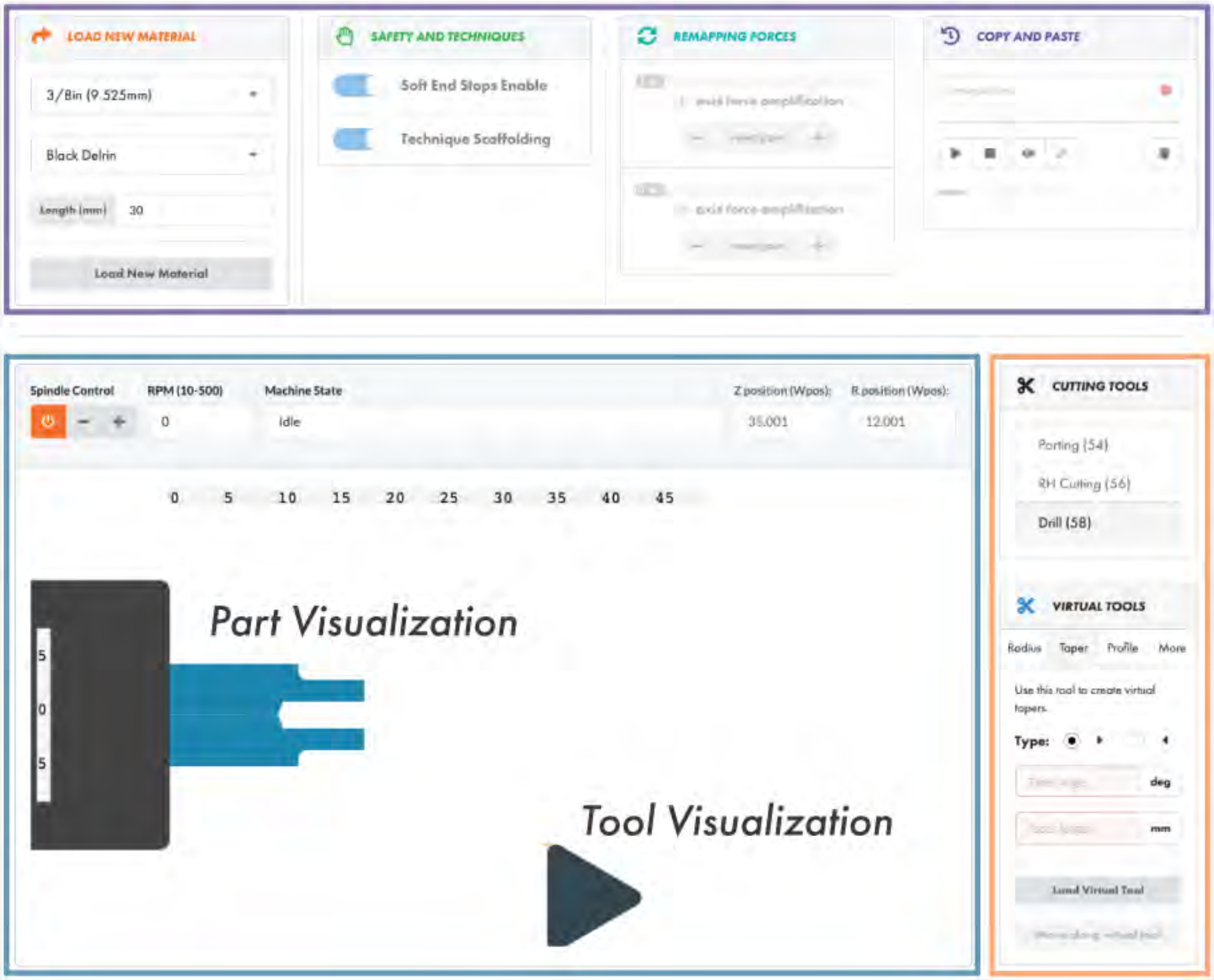
Figure 4.6: Left: User facing UI is rendered in a web browser. Center: Two web servers are used, one to serve the web application, and the other to act as a WebSocket to UART Serial bridge. Right: Three microcontrollers interface with the hardware.

4.4.4 GUI design

The GUI is designed to complement the specialized input modality of the handwheels. While the haptic handwheel user interface is well suited for controlling position and outputting force, not all interactions with the fabrication tool should be expressed through the haptic handwheels. In particular, actions such as selecting a tool to cut, or inputting the size of the stock material are better represented through a standard GUI. The elements of the GUI are visually grouped by function; Figure 4.7 indicates these regions with overlaid colored boxes. At the top of the page (purple outline) is a row of machine settings such as loading new material, or toggling on/off the technique scaffolding. The large pane in the center (blue outline) is a top down 2D visualizer view of the moving tool and component being cut. The geometry of the part being cut is also updated live, using Boolean intersection operations between the stock material and the cutting tool. To the right of this pane (orange outline) is where users select the cutting tools.

4.4.5 Maximizing responsiveness of the system

The end-to-end latency through the software pipeline in Figure 4.6, as measured by the time between sending a command to a handwheel from the front-end JavaScript and receiving the response, is on the order of 10 - 20 milliseconds. During operation, we are able to maintain a frame rate of 40-60FPS. The limiting factor is the recurring draw event for Paper.js, which slows down as more cuts are taken from the rendered polygon. Though this refresh rate is variable, the microcontroller firmware — which controls the haptic control loops — is always running “real-time” under timer interrupts.



3/8in (9.525mm)

Black Delrin

Length (mm) 30

RPM (10-500) 0

Machine State Idle

Z position (Wpos): 35.001

R position (Wpos): 12.001

Use this tool to create virtual tapers.

Figure 4.7: Not all actions can be specified through the haptic handwheels. We use the flexibility of a GUI to complement the handwheel input. Colored lines outline various groups of the UI. Purple (top): tool settings such as toggling on or off haptic scaffolding. Orange (right): interface for choosing cutting tools, both real and virtual. Blue (left): top down 2D visualizer of the lathe, stock material, and cutting tool. Stock material visualization updates as cuts are taken.

Even with low software latency, the overall responsiveness of the handwheel to lathe coupling also depends on the latency of the hardware itself (e.g. the delay in accelerating the lathe's gantry). We additionally adjust our system through the following strategies: First, we tune the acceleration of the stepper motors near the maximum allowable limit, which is bounded by the motors' available torque. Combined with lowering the maximum allowable velocity, these changes allow our lathe to rapidly come up to speed — from stationary to maximum velocity in under 100 milliseconds. Last, because the “transmission ratio” from handwheels to lathe is defined in software, we qualitatively find a satisfying balance point between fine grained movements and maximum lathe velocity. Qualitatively, these strategies result in highly responsive interactions.

4.4.6 System walk-through: adding a guide

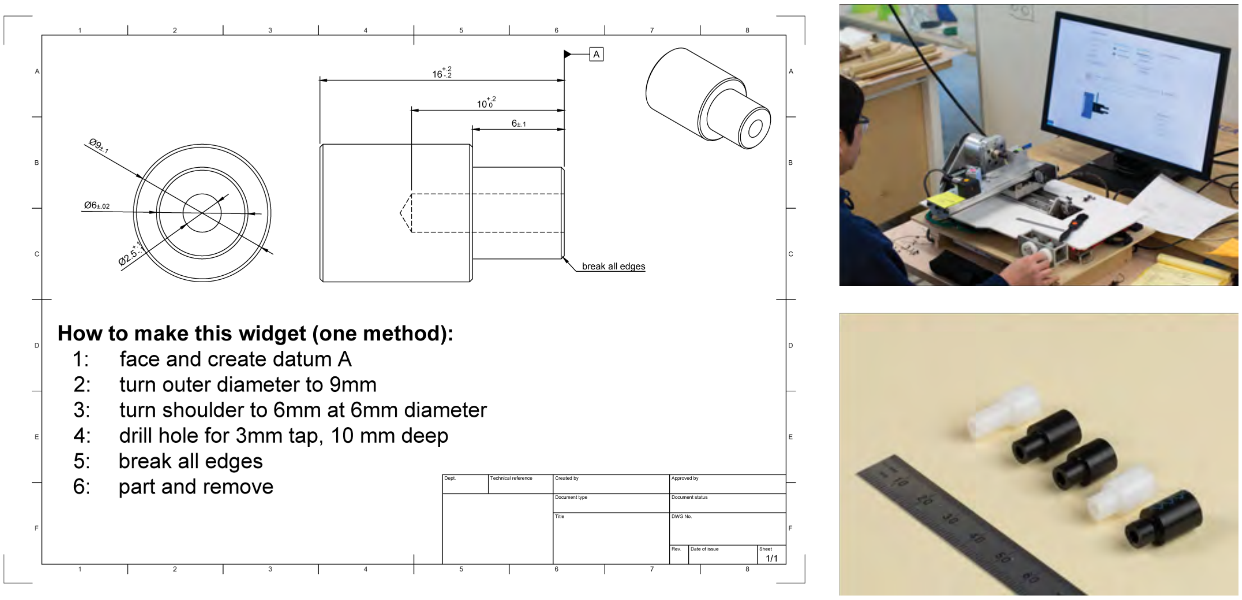
We detail how our system coordinates the different layers of software and hardware to enable the snap-to-guide interaction. Users add guides through drag and drop interactions starting from the rulers located at the top and left of the 2D visualizer canvas. The guide is visualized as a reference line through the GUI and saved on the front-end (Figure 4.7). Simultaneously, the web application issues a command to the corresponding handwheel’s microcontroller to add a virtual detent at that location. This message travels through the WebSocket and is routed to the appropriate hardware serial device by the WebSocket-to-Serial server. Information about the guide location is redundantly stored on both the firmware of the handwheels, as well as in the JavaScript. The handwheel microcontroller uses this position when executing the real-time haptic control loop. Because of the redundant data storage, the microcontroller is not dependent on data from the web application during this low-level control loop.

When the user rotates the handwheels, the corresponding microcontroller streams location update messages to the web application. The web application applies the virtual transmission to these position messages to calculate the desired lathe position. Subsequently, the desired position is sent to the lathe via a G-code command, and is used to update the GUI.

The user feels a centering force through the handwheel when their tool approaches a guide. To achieve this, the guide location is used as the desired position for a simple proportional-derivative (PD) controller by the handwheel microcontroller. As a result, the motor attached to the handwheel is actuated to maintain the handwheel at the desired position. The gains of the PD controller are chosen such that the user will feel the handwheel being pulled toward the guide location, but can insist to move the handwheel away from the desired position. When the user moves the handwheel far enough from the guide, the haptic detent is deactivated. The “feel” of the detent can be modified by tuning the PD controller, which can be interpreted as the properties of the virtual spring.

4.5 User study

We focused our evaluation toward understanding how the modulated haptic feedback might support users during the process of fabrication. Specifically, the user study was designed to probe how each of the individual haptic features, as well as the collective workflow supported by the system, would be used and received.



How to make this widget (one method):

1: face and create datum A
2: turn outer diameter to 9mm
3: turn shoulder to 6mm at 6mm diameter
4: drill hole for 3mm tap, 10 mm deep
5: break all edges
6: part and remove

Figure 4.8: Left: Dimensional drawing given to users during the study. Many of the primary operations possible on a lathe are required to fabricate this component. By design, users also encounter all haptic features while fabricating this component. Top right: User study setup with lathe, handwheels, and monitor to display the GUI. Bottom Right: Widgets made by the five users during the user study. We suggested to the participants to focus primarily on familiarizing themselves with the tool and haptics, rather than achieving perfect dimensions for the outcome.

4.5.1 Participants

We recruited five expert participants (3 male, 2 female) from the staff of an educational makerspace. This targeted sampling was motivated by the unique lens that experienced fabricators can lend. For example, though some of the interactions we evaluate (e.g. technique scaffolding) are especially relevant for novices, expert participants are uniquely positioned to articulate *why* these interactions are or are not useful by contrasting them with their deep experience with manual and CNC tools. As both experienced practitioners and instructors, these participants can also reflect on the types of difficulties faced by novices when they first approach different types of fabrication tools. Lastly, we hypothesized that these experienced users would span a broad spectrum of informed preferences towards fabrication workflows and tools, and could potentially bring these varied opinions to bear when discussing the tool from the perspective of workflows.

Information about users' fabrication background was collected through the recruitment questionnaire. P1 is a fabricator who specializes in industrial and furniture design. P2 is a glass sculpture artist, and P3 is a trained architect and fabricator. P4 is a wood sculpture artist, and P5 is a roboticist who designs and fabricates combat robots. All participants are skilled designers and fabricators with at least ten years of experience in their respective domains. Though all participants incorporate CNC and hand tools in their practice, P1 and P2 lean more towards hand tool usage, while P4 is more CNC tool oriented. P3 and P5 are centrist in terms of preference towards manual or CNC tools. Some of the participants are also responsible for teaching and developing curriculum for CNC machines, such as wood routers and metal milling machines.

4.5.2 Procedure

Users participated in a one-hour study and were compensated with a US$20 gift card. The study began with reviewing and signing research consent documents (10 minutes), followed by an introduction to the various elements of the system and the haptic interactions that are supported (15 minutes). The primary portion of the study was a fabrication task (20 minutes) designed to expose participants to all of the haptic interactions as they turned a component using the system (Figure 4.8 center). After users completed the component, we conducted a semi-structured interview in the remaining time. Following the in person portion of the study, users anonymously completed an online questionnaire.

We tasked all participants with fabricating a specific component. Though we are interested in having future work engage participants in more open ended tasks, we chose this more controlled evaluation first to focus more narrowly on the efficacy of the haptic feedback features. After the system introduction, a printed copy of a dimensional drawing (Figure 4.8 left) and a finished example of the component where shown to the users. The supplied drawing included desired dimensions as well as a suggested ordering of cuts that can be taken to create the geometry. We prompted users to primarily focus on familiarizing themselves with the tool and experiencing the haptic feedback, rather than getting each of the dimensions perfect. Users created the component using acetal, an engineering thermoplastic with good machinability.

The design of this component was inspired by our on campus Mechanical Engineering student shop, where members make a similar component in the final hands-on portion of machine shop training. Many of the primary operations possible on lathes are required, including (1) turning (cutting) the outer diameter, (2) facing (cutting the end of the stock material), (3) drilling, and (4) parting (removing the component from the stock using a thin cutting blade). All three tools currently mounted to our machine are used, across six operations. By design, users encountered all haptic interactions while fabricating this component, for example, setting up virtual guides to drill at the correct depth.

4.6 User study results

All participants completed the component within 15 - 20 minutes (Figure 4.8 right). In the post-task interview, users expressed overall excitement around this type of tool. This result

Figure 4.9: Anonymous questionnaire responses. Left: responses to high level questions around the overall experience. Right: responses to specific haptic interactions mediated by our system.

was corroborated by the anonymous exit survey — participants responded positively to a set of five-point Likert scale questions relating to both the specific interactions as well as the overall experience (Figure 4.9).

4.6.1 Approaches to making the widget

Despite the common end goal, the task allowed for some amount of freedom in how users explored the system. Only P2, the user with the least previous familiarity with lathe tools, followed the ordering of operations suggested by the drawing. Limit stop and technique scaffolding were enabled by default at the beginning of the study, and two participants (P1, P3) disabled this feature when they wanted to make the final "parting" cut closer to the origin, as the software endstops were set conservatively. The feature was engaged during the rest of the study. Feeling the force of cut was also enabled by default for all participants and could not be disabled. Encouraged by the dimensioned drawing, all participants used the snap-to-guide feature extensively to mark locations of interest, such as the outer diameters, and depth of the hole.

4.6.2 User perceptions of haptic interactions

In the anonymous exit survey, users indicated strong positive reactions towards the haptic interactions we implemented (Figure 4.9 right). All participants responded with "Agree" or "Strongly Agree" in response to all haptic feedback features. The haptic snap-to-guide and technique scaffolding were the most and least positively rated features respectively.

Snap-to-guide

Snap-to-guide was an especially compelling feature for many of our users ("Strongly Agree": 4, "Agree": 1). In a very lightweight way, this feature captured one of the main characteristics users look for in CNC tools — automatic accuracy.

> **P5** *[The most memorable part of the tool was] the ability to set the limits so you don't cut too far and actually helps you make the exact right dimensions that you want. That sets it apart from any other machine. Just manual machine or DRO (digital read out) doesn't physically stop you from going past those limits.*

While maintaining the directness of physical fabrication, our tool overlays capabilities typically associated with digital editing. Users also commented on the the satisfying "feel" of snapping to the guide lines.

> **P4** *I want to have an excuse to come back on this tool again because I loved those snap lines and that feeling you get, ... like when you're actually designing in CAD and you snap to align it to the right line and it feels good. Yeah, well this is that physical sensation.*

Scaffolding through haptics

Haptic scaffolding received the least positive feedback in our exit-survey, however all participants were still overall positive: "Strongly Agree": 1, "Agree": 4. We hypothesize that this is related to our users' high baseline familiarity with lathes. P2, the most hand tool oriented participant, uniquely conceptualized our system as a CNC machine to teach material sensibilities.

> **P2** *It's CNC to teach you material consideration... how to feel material.*

Feeling the force of cut

When asked about whether they could feel the difference between cutting "a lot" versus "a little" bit of material, responses ranged from "Agree":3, "Neutral":1, to "Disagree":1. This spread may have resulted from the fact that the force of cut was tuned for the easily machinable plastic used during the user study, in addition to the ambiguity of what "a lot" or "a little" material meant for the user. In the anonymous exit survey, one participant wrote, "When doing actual cutting, especially with the plastic, it felt like butter." In other words, the material chosen may have been too easy to cut to discern any important differences in cutting force.

Despite this, all participants perceived being able to feel the force of cut as a useful feature ("Strongly Agree": 2, "Agree": 3). During the interview, P4 directly commented on the ability to "feel" the machine.

> **P4** *My favorite thing about this so far is that machine feel, [it] is so difficult to communicate to people like what it should feel like.*

Volumes of tacit knowledge [93] are embodied in hand tool usage. While phrases such as tightening something “monkey tight” versus “guerilla tight” try to capture the subtleties of touch, there remains ambiguity that must be calibrated through experimentation and experience. *Turn-by-Wire* short circuit this process by creating an environment in which the feel of a fabrication tool can be communicated directly.

4.6.3 Limitations

While being able to remap forces is what enables many of our system’s interactions, the software mediated nature of the force feedback is also a source of ambiguity. P4 contrasted using our tool with the experience of using a hand plane for woodworking.

> **P4** *If I’m trying to do a hand plane and it’s not cutting, it’s either the way I’ve set it up or my technique cause I’m cutting up the wrong way on the grain. But here there is that kind of black box with a question mark on it in between me and the operation of the tool.*

Another limitation of our implementation is the spatial separation between the visualization of the part being cut, and the physical part itself. During the interview, users commented on wishing that they had more control over the placement of the screen (P5), and having to switch between looking at the screen versus the machine (P2).

4.6.4 User perceptions of workflow

Responses to the workflow were overall positive (Figure 4.9 left). In the anonymous exit survey, no user found the machine “intimidating to use”, and most found it “easy to learn”. While this is likely related to our users’ breadth of fabrication experience, two of our users reported during the interviews that our system drastically contradicted their initial expectations of what learning the tool would be like.

> **P1** *It was a lot more comfortable than I expected because in the beginning you told me it’s kind of like a CNC machine that you guys are building and I thought ... the learning curve is going to be bigger than just what happened today.*

> **P2** *I think the most memorable part was just sort of the ease of use... I mean, honestly I had some hesitation in because my familiarity is very minimal with a metal lathe... I don’t think that is easy to achieve with CNC in terms of making people feel confident almost immediately.*

> **P4** *I love the idea that it’s taking the scariest thing in a typical prototyping shop and making people feel a lot better about using it.*

The haptic feedback in the handwheels give users an additional, and literal, handle to better understand and grasp the system.

4.6.5 Is *Turn-by-Wire* hand tool or CNC tool?

This question prompted diverse responses in our interviews. P1 and P3, reflecting on the direct and handed interactions required to operate it, felt that our system was closer to a hand tool. P5 leaned towards CNC control, noting that under-the-hood, the system is entirely computer controlled and can be programmed to run autonomously without modification. P2, the most hand-tool centric participant, felt that *Turn-by-Wire*is definitely a CNC tool, but "very closely maybe bridges those two... it's as easy to use as a hand tool while embodying the components of a CNC". P4 strongly felt that it was "right down the middle", elaborating that one aspect which prevented the system from feeling entirely like a hand tool is the ambiguity that the by-wire control introduced. Users' varied interpretations of their experiences echo our own intentions for tools like *Turn-by-Wire*, that they occupy a liminal space between hand tools and CNC tools in which users can engage with the capabilities of a CNC machine fluidly and directly.

4.7 Discussion and future work

4.7.1 Combining visual AR with haptic AR

The haptic experiences we develop can be thought of as a kind of augmented reality — the sensations a user feels while cutting are curated and augmented, instead of a literal portrayal of the induced forces. We envision blending these haptic experience with visual augmented reality. This combination can be used to build rich experiences for scaffolding, for example to render both the look and feel of cutting difficult geometries or materials. For maximum safety, entirely virtual materials can be "loaded into" the lathe for experimentation and practice (Figure 4.10). Softer test materials can also be made to feel like those that are more difficult to machine, to reduce the amount of graphic rendering required to create an immersive experience. From a more practical standpoint, an AR visual display would also address the physical separation in the current system between the lathe and digital UI. Hands-free AR interfaces can be implemented through traditional head-mounted devices or fixed tablet computers. Tablets in particular can be adapted to the form factor of a chip guard, which is typically a piece of clear plastic that shields users from cutting debris.

4.7.2 Experiencing how cuts shouldn't feel

Many of the haptic interactions we introduce in this paper focused on how this modality can be leveraged to ensure that the user is operating the machine safely. Similarly, haptics can also be utilized to show users how certain operations should *not* feel. Whether through a jittery handwheel, rendering rough textures, or kicking back, the haptics could potentially be leveraged to reflect the tool's own uncertainty about what the user is about to do, or used during training to familiarize users with the limitation of the machine.

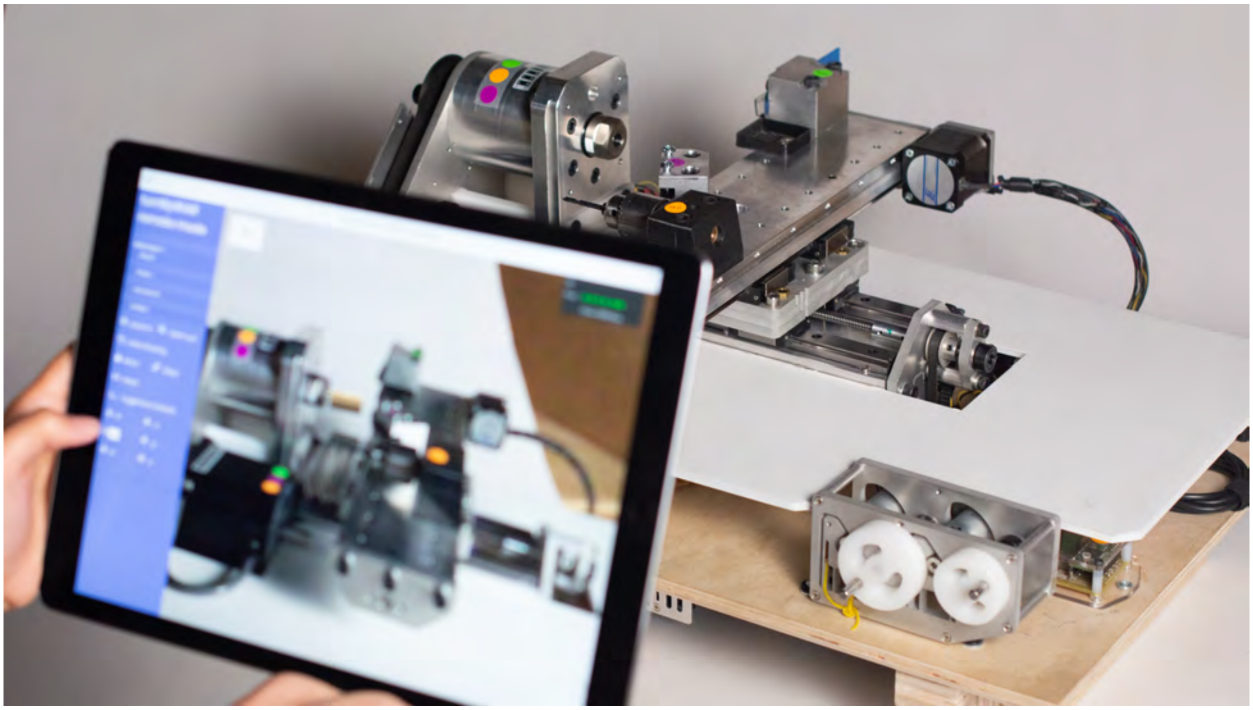


Figure 4.10: In this AR prototype, a tablet computer renders a virtual wood material into the lathe (round, tan colored cylinder on screen). Apple's ARKit and the tablet's sensors are used to orient the rendering of the virtual material. More details can be found in [102].

4.7.3 Force feedback from lathe motors

Both handwheel and lathe motor mounts have been designed such that the output torque of the motor can be directly measured through displacement sensors (such as strain gauges). This capability can be used to more accurately convey cutting forces to the user through control loops that measure and output specific forces. This is especially important when working with anisotropic materials such as wood, which respond very differently to cutting tools depending on the orientation of the grain. Time-varying parameters such as tool wear could also be captured and conveyed.

4.8 Conclusion

Turn-by-Wire utilized haptic feedback handwheels to mediate the hands-on fabrication process, in terms of both feedback and control. Leveraging the "by-wire" control fully, I developed haptic interactions for feeling the location of a virtual guide, or for learning the proper technique for using a particular tool. Notably, these detents is not real outside which is not a real force that exists outside of software, but nonetheless facilitates the process of fabrication. Five expert participants, whose experiences span diverse workflows and domains, used *Turn-by-Wire* and responded positively to the haptic interactions.

A lathe was chosen as a case study to concretize ideas around direct yet mediated control, but the underlying ideas that guided this work can be broadly applied to other fabrication tasks. Many machine shop tools — milling machines, drill presses, grinding machines — are also controlled through manual handwheels, and readily lend themselves to augmentation through similar systems. However, to generalize these ideas to a wider range of tools, an alternative approach would be to augment the user themselves, rather than to augment each additional tool individually. In the final project, I discuss how this goal can be achieved through a novel use of industrial robots.

Chapter 5

aDroid

In this chapter, I explore how the ideas embedded in *Turn-by-Wire* can be generalized beyond a single tool. How can the capabilities of digital editing be broadly incorporated within hands-on fabrication? One approach is to create a new system for each additional tool that could benefit from this style of interaction. This piecewise approach is completely sound. As we saw with *MatchSticks*, there are many potential benefits to working with a focused, context specific tool. However, a complementary strategy is to directly augment the user, rather than any particular tool. This approach may be more extensible — if the same hardware system can allow a user to interact with *any* tool "by–wire", then augmenting an additional tool becomes simply a matter of software, rather than designing and creating an entirely new system of sensing and actuation. In this chapter, I explore how an industrial robot can be applied to this latter approach.

aDroid, a play on "adroit" and "a droid", is a single robot which lends precision and accuracy to users working with hand-held tools. When a tool, such as a drill, is mounted to the robot, the user can hold and move the tool directly; backdrivability is achieved through force sensing and software. However, depending on the tool and scenario, the system can selectively restrict certain motions. In the resulting interaction, the robot acts like a virtual 'jig' which constrains the tools motion — augmenting the user's accuracy, technique, and strength, while not diminishing their agency during open ended fabrication tasks. I complement these touch based interactions with projected augmented reality for visual feedback about the state of the system. I show how tools augmented by *aDroid* can support hands-on fabrication, and discuss how it can be configured to support other tasks within and beyond digital fabrication.

5.1 Introduction

Industrial robotic arms are powerful, accurate, and versatile tools. From freeform 3D printing [23, 89] to timber frame manufacturing [108], researchers have explored numerous creative applications that leverage the capabilities of these machines. Current digital fabrication

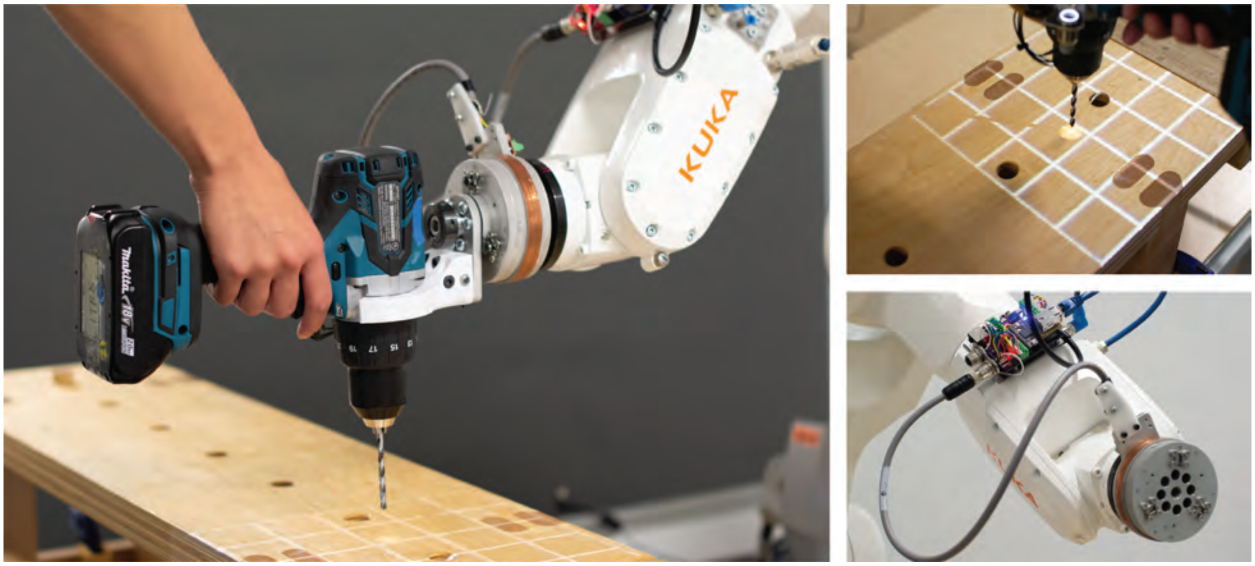


Figure 5.1: Left: *aDroid* allows users and robots to work collaboratively in workshop environments. Tools mounted to the robot can be moved directly by the user, but the robot computationally constrains the motions to suit the tool and context. Left: The system is currently set up to assist a user with a drill. Right top: A projection AR system communicates information about the current tool, such as its position and orientation with respect to the workpiece. Right bottom: A custom force-torque sensor senses how the user wants to move the robot.

workflows for industrial robots share many similarities with the workflows of more commonplace digital fabrication tools, such as 3D printers or CNC routers: a desired geometry is designed in computer aided design (CAD) software, the corresponding toolpath is developed in a computer aided manufacturing (CAM) package, and the machine executes the toolpath to physically instantiate the desired geometry [112]. Similar workflows yield similar limitations: namely, there is high overhead for defining a desired geometry, the final toolpath must be perfect before the user hits “print”, and the machine does not adapt to any deviations between the real world fabrication task and what was planned in the simulated environment.

How might users gain the benefits of working with digital fabrication tools without being limited by existing workflows? The previous projects addressed this question from the perspectives of context specific tools. Though these ideas can be generalized through additional context specific tools, they can also be generalized in more interesting ways through a general purpose tool: industrial robots.

In *aDroid*, we seek to support users during hands-on fabrication with a system that allows them to rapidly borrow accuracy and precision from robotic arm as needed. The high-level approach is inspired by how jigs and fixtures support users by selectively constraining the many degrees of freedom in our bodies. For example, a drill press constrains the motions of a drill along a line, and a miter box constrains a hand saw to move accurately within a plane. Similarly in *aDroid*, no motions are pre-programmed, only constraints — as the user tries to move the tool, motions are generated on the fly as the system responds to the users applied forces and the current constraint. In effect, the robot becomes a haptic exoskeleton whose compliance can be computationally tailored to the task at hand. These haptic interactions are complemented by a projection augmented reality interface which offers in-situ visual feedback about the current tool, such as its position with respect to a workpiece (Figure 3.1 right).

As with physical jigs, our system is useful not because it generates or executes motions, but because it defines what motions are allowable. Likewise, these jigs do not compel users to fabricate a particular object; instead, users reconfigure jigs as needed to support their making process. *aDroid* however, takes full advantage of the digital medium. Consider some of the opportunities afforded by a computationally generated "drill press": First, the "drill press" can be downloaded and shared. Next, settings for the "drill press', such as the drilling angle or depth limit, can be instantly and accurately set, saved, or restored. The "drill press" not only constrains the motion of the tool, it can also mediate forces, generating sensations such as real world snap-to-grid with haptic feedback. Because all interactions are mediated through software, the "drill press" directly knows where and how holes have been created. Moreover, this "drill press" can be reconfigured to become any other jig through software.

Collectively, this system allows us to begin incorporating capabilities we typically associate only with digital authoring directly within a hands-on making environment. Primarily, this work contributes a system embodying the concept of virtual jigs and fixtures for hands-on fabrication. To demonstrate a range of applications, we showcase four tools that can be used with *aDroid*, each of which highlight a specific facet of the system. In addition, to begin to address the prohibitive cost of robots and their accessories, we detail the design and implementation of a low-cost force-torque sensor which is central to the types of interactions we present.

5.2 Related work

Augmenting hand-held tools have been explored in digital fabrication, robotic surgery, construction tools, vehicle control, and more. Though different in context, works across these domains share a similar framing — greater strength, accuracy, or performance are desired, without sacrificing the unique capabilities afforded by direct hands-on control. Similar to *MatchSticks* and *Turn-by-Wire*, *aDroid* considers how computation can support the process of fabrication. However, rather than focus on one domain, *aDroid* considers how, from both interaction and technical perspectives, a general purpose system can be imbued with the capabilities explored in the two previous projects.

5.2.1 Virtual and physical fixtures

Ideas around virtual fixturing were explored by early telerobotics researchers. In 1993, Rosenberg demonstrated a system in which a user controls a robot using a haptic controller to complete a peg-in-hole insertion task, and various virtual constraints – communicated through haptic feedback – assist the user with spatial orientation [100]. In a similar vein, the original instantiation of “Cobots” was unable to move on its own [90]. Instead, it simply guided an operator’s motions by adjusting a steering mechanism that defined a path of least resistance. Another early system adapted this concept for revolute joints; motors where replaced with directional clutches [124]. While these techniques have been investigated within the human robot interaction and surgical robotic communities, we apply the concept of virtual fixtures for mediating physical tool usage in digital fabrication.

Physical fixtures are widely used in hands on fabrication as a means to improve the accuracy and repeatability of hand tools. For woodworking alone, a quick glance at tool retailer catalogues from Rockler or Lee Valley will reveal hundreds of different jigs that each selectively constrain the motion of a particular tool in a particular way. For example, miter boxes constrain a hand saw to move along a plane, drill guides constrain drills to move along a line, box jigs constrain routers to make successive cuts with consistent spacing. Researchers within HCI have also explored how woodworking jigs can be computationally designed and fabricated [65]. Beyond woodworking, HCI researchers have examined the role of computational jigs in wire [123] and fiber [143] wrapping. While our system can store jigs in software and ‘render’ them on demand, this is only part of our contribution. Most importantly, we leverage the concept of a virtual fixture as a means to incorporate digital editing affordances directly within physical fabrication.

5.2.2 Controlling industrial robots today

At the lowest level, robot motions can be defined in manufacturer specific programming languages such as Kuka’s KRL or ABB’s RAPID. These languages are essentially more sophisticated versions of G-code, and allow for looping, branching, and GPIO control. When industrial robots are used in digital fabrication, their workflows closely resemble that of other digital fabrication tools, with the primary difference being the choice of post-processor [112]. Spurred by developments in film-making [14], animation tools like Autodesk Maya have also been co-opted as a way to simulate and generate robotic motions [71]. However, toolpaths generated through these external tools are often static — the robot ignores discrepancies between the simulated and real environment.

When adding interactivity to industrial robot motions, one approach is to offload computation away from the low level, vendor specific programming language of the robot by exposing an interface for controlling robot motions [36, 106, 38]. External computation then coordinates robot motions based on additional sensors, allowing for motions to dynamically adapt. Using these interfaces, researchers have explored algorithms for updating robot trajectories based on sensor input, rather than relying on a pre-planned path. Reflexxes [55] and Quipt [35] are two notable examples from engineering and art practice. In contrast, our work sidesteps toolpath generation through the use of virtual fixtures. No motions are predefined, only constraints — toolpaths are generated on the fly as users exert forces on the robot.

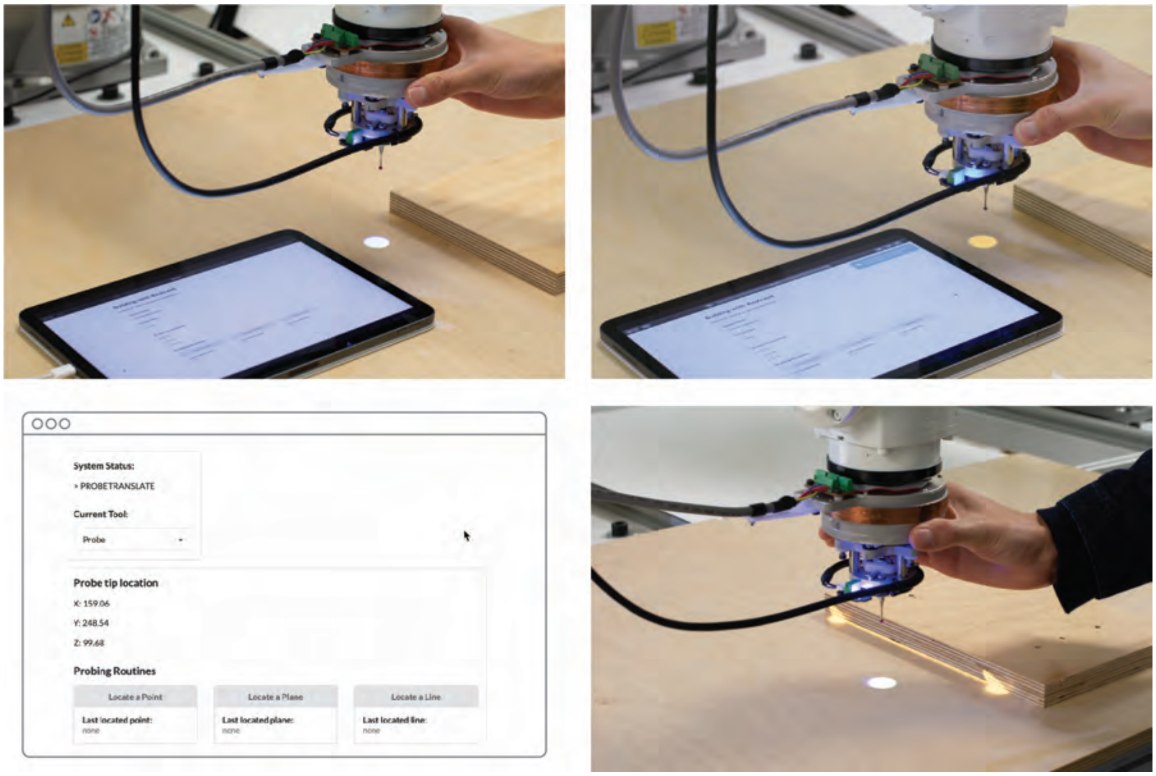
5.2.3 Robots in fabrication and haptics

Within digital fabrication, robots have been used as a platform to support 3D modeling with fast physical feedback [89, 76], as well as multi-axis toolpaths for 3D printing [23, 47]. Segments of a large scale fabrication task were delegated to human and robot workers in [58]. Within the architecture research community, industrial robotic arms are widely used as a platform for exploring new manufacturing materials and workflows [119, 94]. One notable example is the ICD/ITKE Research Pavilion 2014-2015, in which robot arm was used to lay carbon fiber onto the inside of an inflated plastic sphere. In this project, researchers addressed the unique challenge of defining the robot's motions based on both the design of the structure as well as the shifting shape of the inflated form; the later was measured through force feedback at the robot end effector [129]. Another is *Adaptive Robotic Carving*, in which the authors recorded and analyzed the motions of a craftsperson working with a powered reciprocating gouge [16]. A wide range of data—collected through motion capture, force sensing, digital scans of the resulting geometry—was analyzed with machine learning, and used to create toolpaths for a robot equipped with the same gouge. In this project, researchers sought to distill the tacit knowledge embodied in the motions of the craftsperson, such that this knowledge can be utilized to operate an autonomous system. Though *aDroid* and [16] share many similar underlying technologies, these technologies are leveraged towards diverging goals. In the building construction industry, early startups and industry veterans alike are mounting industrial robots to mobile carts in order to automate, or partially automate, tasks such as as overhead drilling [48], drywall finishing [17], and general purpose fabrication [9].

Physical contact between humans and robots is often viewed as a risk to be mitigated. However, some robot manufacturers in industry have begun to incorporate force-torque sensing, which allow users to "program" robot positions by directly moving the robot [56, 62]. During these types of interactions, the robot can also communicate additional information through touch, such as repelling forces near singularity configurations [27]. In the context of VR, researchers have also explored how a robot can be used as a general purpose haptic controller for rendering force feedback [26]. Rather than use physical contact only for the purpose of demonstrating a path that is ultimately meant to be performed autonomously by the robot, our system utilizes continuous physical contact to mediate the direct and collaborative use of hand-held tools.

5.3 Interacting with *aDroid*

With *aDroid*, users directly control—through touch—tools mounted to an industrial robot. This primary interaction mode is enabled by our force-torque sensor and software stack; the system measures the forces the user exerts on the tool and commands the robot to move accordingly. In addition to the primary touch based interactions, we also provide standard user interface elements for visual feedback and user input. We display spatial information about the systems state (such as a tool's current position) in-situ using a projector mounted to a tripod. For actions such as adjusting settings for a tool, the user interacts with a GUI on a tablet computer. We envision that these tablet interactions can be unified with the AR system, but in our current implementation, we use a tablet to ease how user inputs are captured.



System Status:
› PROBETRANSLATE
Current Tool:
Probe

Probe tip location
X: 159.06
Y: 248.54
Z: 99.68

Probing Routines

Last located point: none
Last located plane: none
Last located line: none

Figure 5.2: A: To ready the probe, the user selects the tool from the graphical user interface, and follows the on-screen instructions for mounting the tool. B: When the tip of the probe contacts an object, the probed location is recorded by the system, and is highlighted by the projection AR display in orange. C: Using the GUI, the user can set up the system to measure specific geometries, such as edges. D: To find an edge, the GUI prompts the user to probe two points along that edge. The AR system displays the probed locations, as well as the found edge shown in orange.

To demonstrate a range of applications, we showcase four prototype tools that can be used with *aDroid*. Each tool highlights a specific facet of the system. With the measurement probe, we show how the initial correspondence between the physical and digital models can be established. With the track saw, we show the system can record a user's fabrication process. With the drill, we show how accuracy can be borrowed from an industrial robot during manual workflows. And with the camera rig, we show how users can define custom constraints to tailor the system for their needs.

Measurement Probe — The measurement probe is used to accurately find the stock material (Figure 5.2). After the tool has been mounted, the user can move it about the workspace by guiding it directly through touch. The force-torque sensor located between the robot and the probe senses the applied forces, and our software moves the robot in response. When the probe tip touches an object, the robot retracts the probe tool to a safe position, and the system records the probed position. Using the probe, the user can rapidly and accurately measure the size and shape of the stock material. Unlike in traditional CNC workflows, our user is not locating the material in order to align it to a predefined toolpath. Instead, she is initializing an environment in which the act of physically fabricating an object is simultaneously the process for generating that object's digital model.

Track Saw — A track saw is a tool in which the user guides a circular saw along a track. In this scenario, the robot holds the intermediary tool—the track—rather than the saw itself. As the user moves the track, the AR system displays where the cut will occur relative to the edge of the track (Figure 5.3A). When the user is ready to make a cut, she locks the track by pressing a 'lock' button on the track (Figure 5.3B). We incorporated this button on the tool itself to minimize the need for a user to switch between the GUI and the tool for frequently used inputs. In this interaction, the user can freely work with the tool to choose how and what to cut. Because the track is constrained by the robot, all cuts that are intended to be parallel remain parallel, and angled cuts can be precisely positioned. In addition, because all interactions are mediated through the system, the sequence of cuts our user makes are known, and a record of how the workpiece was modified is generated. In other words, when our user interacts with our system to physically fabricate an artifact in the real world, she is simultaneously creating a digital model of that object (Figure 5.3E).

Drill — When the drill is loaded, the AR system projects a grid onto the workpiece, along with crosshairs to indicate where the drill is pointing (Fig. 5.4A). To assist with precisely positioning holes, snap-to-grid is implemented physically (Fig. 5.4B). Physical buttons next to the drill are used to toggle between controlling translation and orientation, as well as to for drilling (Fig. 5.4C). While drilling, the system will only allow motion along the current axis of the drill bit (Fig. 5.4E). With *aDroid*, the user is able to leverage the accuracy of computer controlled manufacturing tools, but this accuracy is made accessible without requiring the user to numerically describe every aspect of controlling the drill: where to drill, how fast to drill, or how these settings should change depending on how the cut is progressing. Instead, the system instantiates an environment that allows the user to "just use a drill", while being supported by the accuracy of a digital fabrication system—fluency with traditional computer aided manufacturing software is not required.

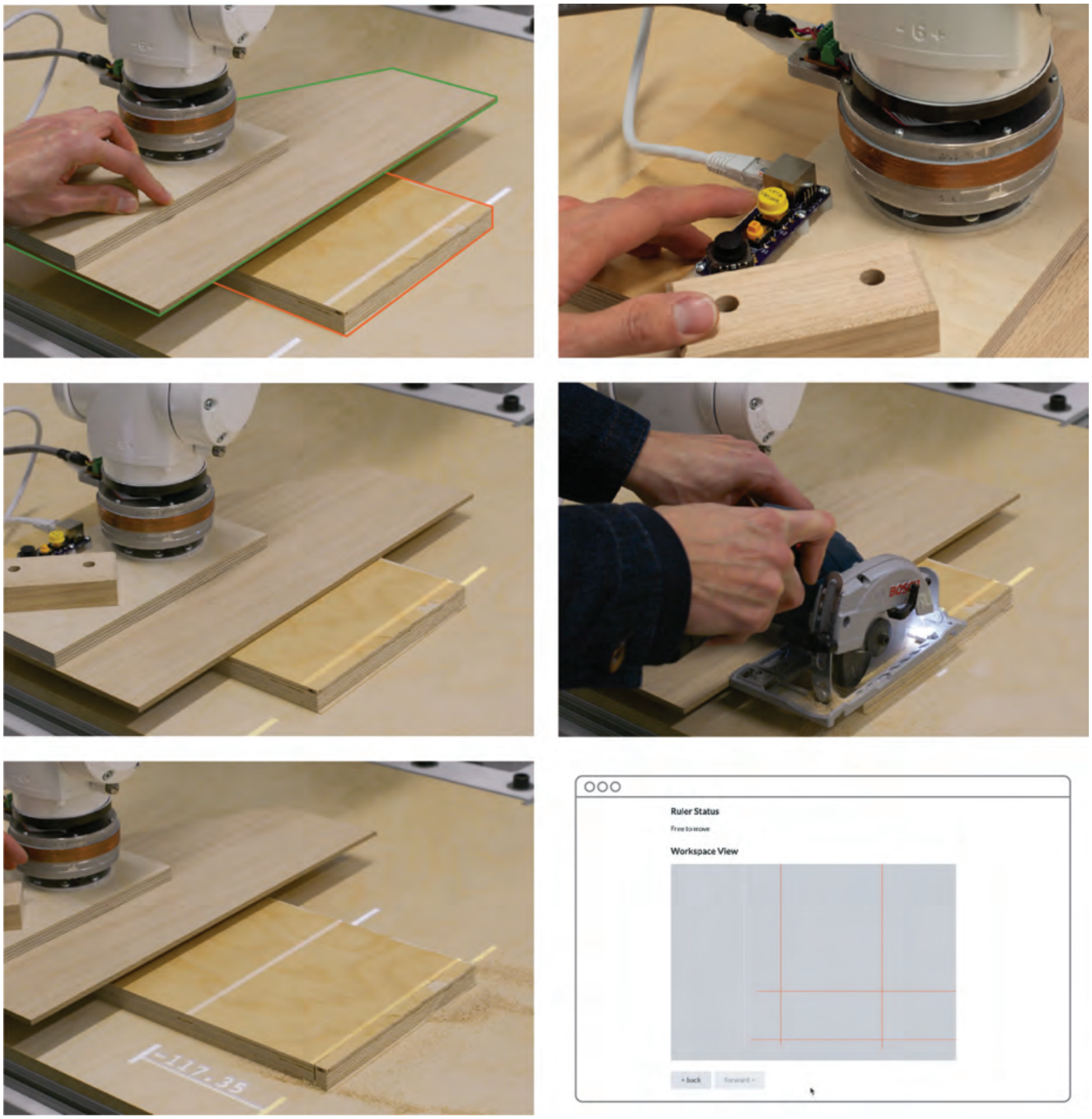


Figure 5.3: Track saws are commonly used to make straight cuts on planar materials. A: With the track saw, the robot holds an intermediary tool, the track. The AR system projects a white line to visualize where the saw will cut relative to the track. For clarity, the track has been outlined in green, and the workpiece in orange. B: A physical 'lock' button on the track allows the user to lock and unlock the track. C: When the button is pressed, the robot lowers the track to the height of the stock material The AR visualizer toggles the saw line color from white to orange. D: With the track locked, the user can proceed with their cut. E: The distance from the last cut is displayed. F: In the GUI, the user can view and navigate the history of previous cuts, like a lightweight CAD model.

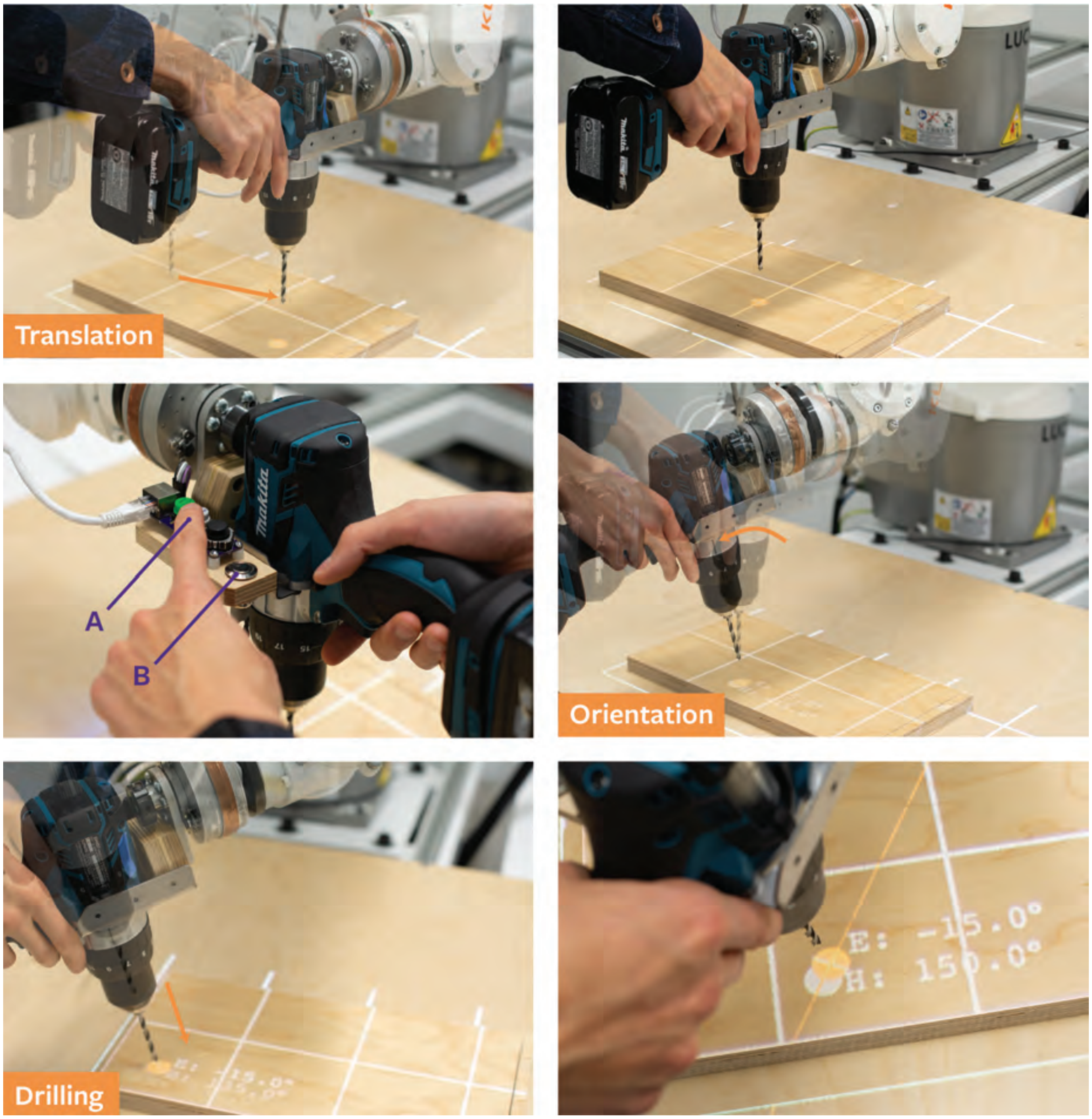


Figure 5.4: A: For the drill, the AR system projects a grid and crosshairs onto the surface of the table. B: When the user moves the drill near a grid line, the drill will gently snap into the position of the grid. The current grid line is illuminated when the drill is "snapped-to" that line. C: Physical buttons on the drill allow users to toggle between translation (A) and orientation (D), as well as to ready the system for drilling (E). E: While drilling, the robot constrains the drill to only move along the axis of the drill bit. F: When the user is controlling the orientation, the system visualizes the current heading and elevation.

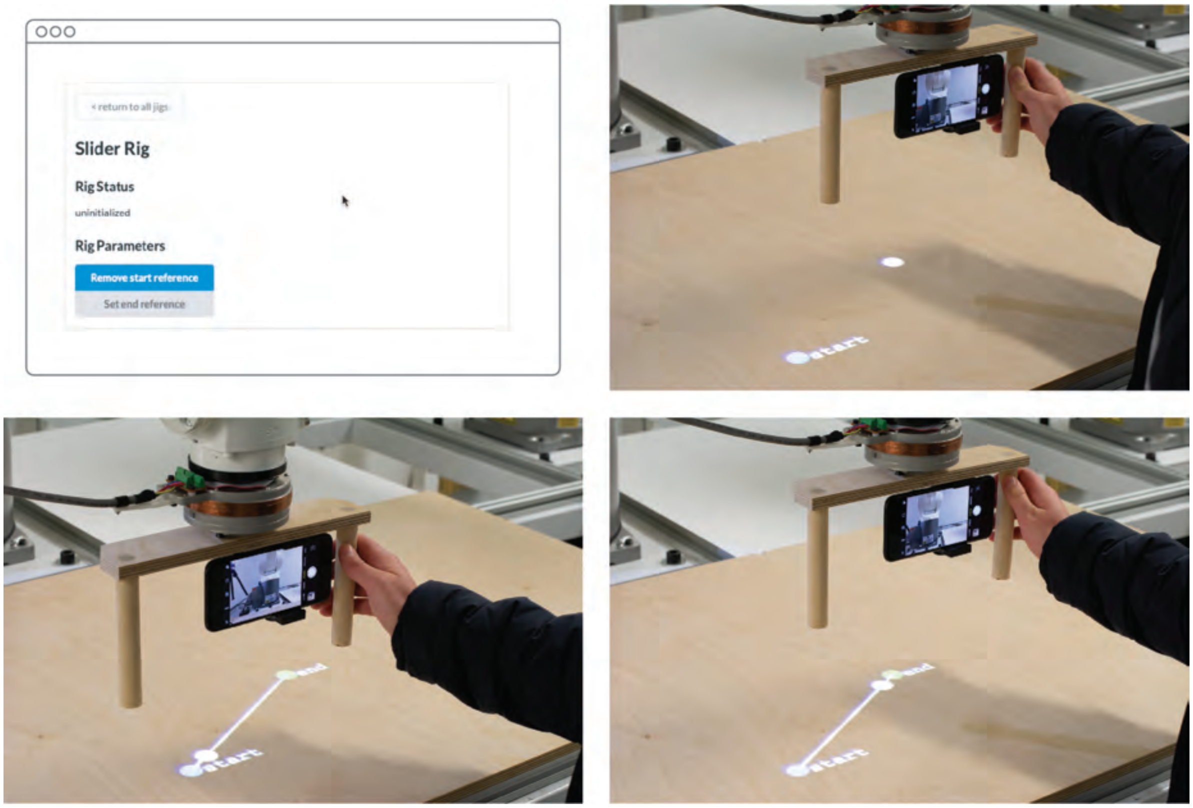


Figure 5.5: Camera rigs can also be thought of as mechanisms which selectively constrain the motion of a tool. Here, the user defines and uses a virtual slider. A: After moving the camera to a desired start position, the user sets that position using the GUI. B: The projected augmented reality view updates with the position of the configured start location (blue), as well as the current location of the camera (white). C and D: After setting the end position, the slider is ready for use.

Camera Rig — Documenting process is an important aspect of many creative practices. Though not strictly related to fabrication, photography and videography equipment such as tripods, sliders, and cranes can be similarly recast as jigs and fixtures—mechanisms which selectively restrict the movement of a tool. *aDroid* can adapt itself to become many of these physical jigs through software. For example, the system can be adapted to become a camera slider (Figure 5.5). As with the other tools, the user can move the camera freely; using the GUI, she can indicate if the current position should be set as the start or end location of the slider. After the user has defined these parameters, she can freely experiments with camera positions and compositions, while the system enforces the constraint that she has defined (Figure 5.5 right). This example highlights how *aDroid* allows the user to create a custom constraint for their particular task.

5.3.1 Proposed scenario: building a step stool

Figure 5.6: In a proposed fabrication scenario, we walk through how a user leverages *aDroid* to create this step stool.

To contextualize how these tools can be leveraged during a fabrication workflow, we walk through a scenario in which our user is building a wooden step stool: For a design with only a few components (also a design that she currently needs one copy of), she works directly with the tools and materials available at hand rather than draft the entire design first on a computer. First, she sketches out a design in which the legs are angled away from the center of the seat (Figure 5.6 left). Rummaging through her scrap bin, she finds a piece of wood that is suitable for the seat, as well as some dowels for the legs. After mounting the piece to her workbench, she uses the probe to measure the size and position of the material – the initial geometry is measured, and subsequent operations can be contextualized. To shape the wood for the seat, she uses the track saw. She is able to work directly and immediately with a computationally augmented tool. No CAD model is required; rather, the process of working physically with the tool generates the CAD model. After the overall size of the seat has been cut, she removes the track from the robot and readies the drill. She quickly consults her sketches, orients the drill at the desired angles, and begins drilling holes for the legs. Unlike a manual drill press, software can help ensure that the setup for this operation is quick and correct. Without the aid of the robot, she uses a lathe to turn the ends of the legs to interface with the holes on the seat. The stool is almost complete! She arranges the components of her stool onto her workbench, and mounts her smartphone to the camera rig. She defines a virtual slider, and composes a few artistic videos of her knolled[1] components to document her process.

---

[1] https://en.wikipedia.org/wiki/Tom_Sachs#Knolling

5.4 *aDroid* architecture

The hardware of our system primarily consists of a robot, a custom designed force-torque sensor, and the end effector tooling. The computation is handled by a server which interfaces with the hardware devices, as well as the front-end user interface. We will discuss the design of these components in detail, and highlight elements of the design as they relate to goals such as safety, responsiveness, and modularity.

5.4.1 Interactive control of an industrial robot

In the context of accurately and repeatably fabricating physical objects, the stiffness and accuracy of industrial robots is a major asset. We use a Kuka Agilus KR6-R900, an industrial robot with a maximum reach of 900 millimeters, a maximum payload of six kilograms, and a pose repeatability of +/- 0.02 millimeters. This robot can be operated through the KRL programming language. In order to interface the robot to external computation, the user can purchase additional software upgrades from Kuka such as MxAutomation or EthernetKRL, each of which cost thousands of USD. One of the few free options available for communicating interactively with Kuka robots is KukaVarProxy (KVP) [2]. This program establishes a server on the robot control computer, and allows clients to read and write global variables across a TCP/IP connection.

To communicate motion commands to the robot, we define a series of control and position variables accessible by both KVP server and the current Kuka Robot Language script. Our KRL script continuously checks the status of these variables, and commands the robot to move when appropriate. In our initial test, the motions generated through this scheme were jerky — the robot would come to a complete stop after completing each successive motion. To improve the robots responsiveness, we implemented a queue in KRL with additional control variables. This yielded significantly smoother motions; the robot's motion planner is able to plan the velocities of multiple moves in advance rather than coming to a complete stop after each move.

One disadvantage of using a robot like the Kuka KR6 is that they are not inherently safe: they are heavy, can move quickly, and are not backdrivable. we have taken many precautions within both hardware and software to ensure the safety of the operator in these interactions. First, the maximum speed of the robot is limited at 250mm/s; no interface is exposed for increasing this velocity. Second, a safety button on the teach-pendant of the robot must be depressed at all times. If the button is released, or pressed too hard, the robot will come to a controlled stop. If the user is operating the robot by themselves, one hand will be occupied by the teach-pendant. To allow the use of both hands, we envision using a foot pedal perform a similar safety check.

---

[2] https://github.com/ImtsSrl/KUKAVARPROXY

5.4.2 Force-torque sensing

A force-torque sensor is central to the interactions we discuss in this paper. However, commercial force-torque sensors can cost upwards of \$5000 USD, which is nearly 20% the cost of the entire robot. We endeavored to build a sensor that was high resolution, easy to manufacture, and low cost. The prototype sensor was fabricated at an academic makerspace for under \$100 USD, and measures two axis of torque (X and Y) and one axis of force (Z). We characterized the Z-axis performance and measured a signal noise (standard deviation of the signal over 100 samples) of 0.1 Newton. This resolution is comparable to existing commercial force-torque sensors today.

Like most force torque sensors, this sensor measures forces and torques via elastic deformations. Deformations are controlled through mechanical flexures and sensed using an array of printed circuit board (PCB) spiral inductors. The effective inductance of the PCB inductors changes as a function of distance to nearby conductors; inductance shifts are subsequently measured using off-the-shelf 28-bit inductance to digital converters ICs. We developed this sensor only as far as to support the interactions we describe in this paper, and additional refinement and characterizations are possible. More details about this sensor's design, construction, and evaluation can be found in Appendix A.

5.4.3 End effector tooling and electronics

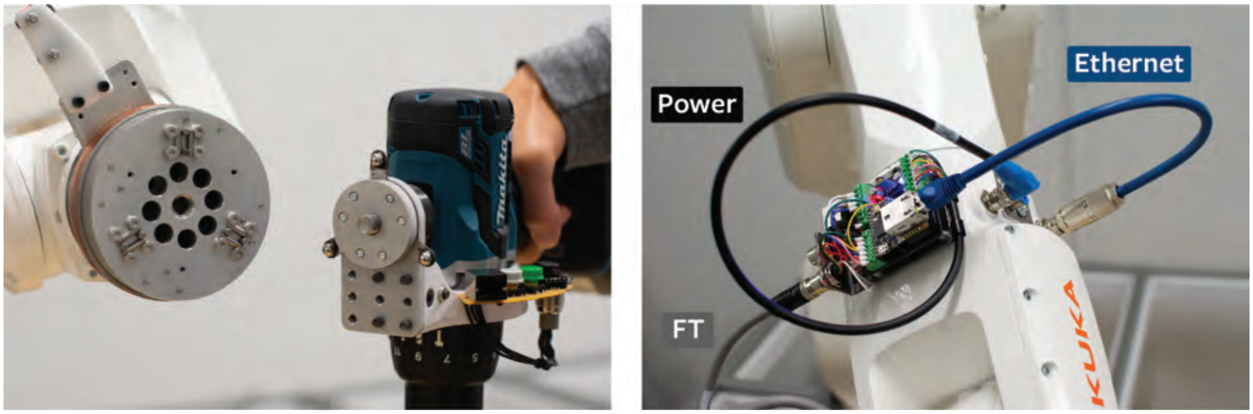


Figure 5.7: Left: The force-torque sensor is attached to the end of the robot arm. The kinematic coupling on the force-torque sensor allows for tools to be easily and repeatably mounted. Right: Electronics on the end effector communicate information about the force torque sensor and the currently mounted tool through an Ethernet cable routed through the body of the robot.

When designing the end effector interface, we wanted to create a way for users to quickly and accurately switch between different tools. The default mechanical interface at the end of the Kuka robot is a flange with eight screws, which we use to attach the force-torque sensor (Figure 5.7 left). A kinematic coupling, chosen for its high positioning repeatability [113], is used to mount various tools to the force-torque sensor. As a result, the position and orientation of tools do not have to be recalibrated if they are removed and remounted. In addition, users can attach a tool to the robot using a single large M10 thumb screw. Again, this mechanism is designed to be easily fabricated, with the machined grooves of a typical kinematic coupling replaced with an assembly of dowel pins. While most kinematic couplings are not suitable for high loads, the ball bearings of this kinematic coupling are mounted to flexures. The flexures bend as the kinematic coupling is tightened, so that forces are not carried by the kinematic coupling itself.

As with the drill, end effector tools may have addition inputs and outputs that need to be monitored by our system. To communicate with these devices, we chose to bypass the GPIO monitoring available within the Kuka robot language because the IO at the robot's wrist is relatively limited (4 inputs and 2 output). In addition, we wanted to offload computation away from the low level programming language of the robot controller itself. The robot includes an interally routed ethernet cable. We mount a microcontroller to the robot's wrist, which communicates to our main server as a UDP server using this physical connector (Figure 5.7 right). This microcontroller is used to communicate to the simple end effector user interfaces, as well as the force torque sensor.

Next, I highlight implementation challenges and details specific to each end effector tool:
The touch probe's tip is mounted to a spring-loaded kinematic coupling, which allows it to return consistently to the same location after it has been displaced (Figure 5.8 left). In this design, the kinematic coupling itself is configured as three rudimentary electronic switches. When the coupling is fully engaged, the circuit is complete; when the probe is ''tripped", the coupling disengages, disconnecting the circuit. We connect this electronic signal to the Kuka's GPIO rather than our end effector electronics interface. This allows us to use interrupts on the robot controller to directly monitor the probed position.
The ruler is connected to the robot with a flexural plate in series with our quick-connect adapter (Figure 5.8 middle left). We design the shape of the flexural plate such that it can accommodate out of-plane-deflections (e.g. working with materials that are less than perfectly planar), but rigid within the plane to ensure accurate cuts.
The drill tool consists of an off-the-shelf drill attached to our mounting bracket. We selected a drill that has an attachment point for a side handle, and initially designed the bracket to grip the drill where the side-handle would normally attach. However, because of how short the gripping area was, the tool could rock slightly. We revised the design and fixed the drill using the existing fasteners that attach the drill's gearbox to its body.
The camera end effector is inspired by existing camera rig designs. For simplicity, we use a smartphone as the camera and utilize an off the shelf phone mount to attach it to the rig.

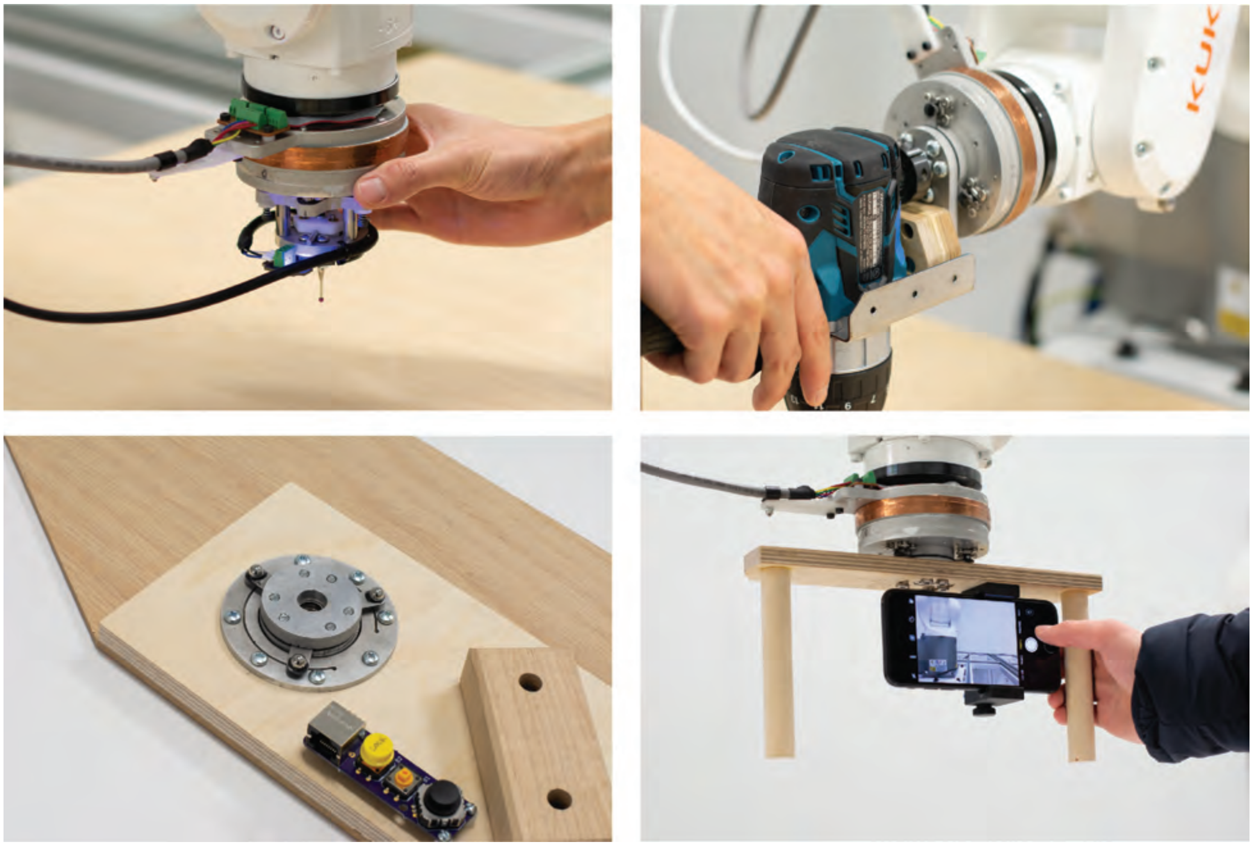


Figure 5.8: A. The probe uses the contact points of a kinematic coupling as electromechanical switches for detecting when the probe tip has been displaced. B. The drill is mounted to the bracket using the fasteners that secure the drill's gearbox to its body. C. The track is attached to the robot through a flexure, which is designed to compliant out of the plane, but rigid within the plane. D. The camera uses an off the shelf smartphone mount.

5.4.4 Choreographing interactions

We establish one primary server to connect to the robot, external hardware, and front-end user interfaces. Within the server, software modules are created for interfacing with each of the hardware devices, and communication between objects is done through a publisher-subscriber event bus. In addition, the server maintains a state machine that tracks the current tool, active constraint, and workspace configuration. When the state changes (for example, when a user mounts a drill) the server configures a new callback function to listen for force torque updates and calculate and emit the desired robot motion request. The robot is not inherently backdrivable, and the compliance is defined entirely in software. When fixtures are in use, the callback function is configured to only allow motions within the fixture's unconstrained degrees of freedom.

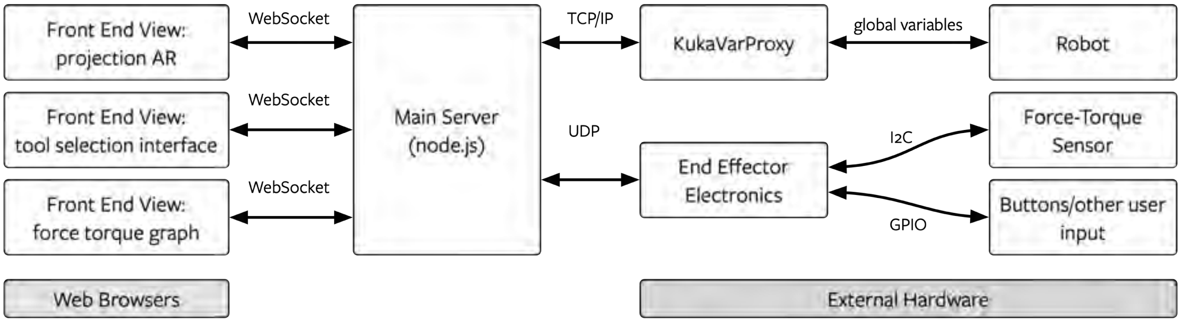


Figure 5.9: The interactions are primarily controlled by a server communicating to each of the hardware devices (right), as well as the user interface elements (left).

All front end user interfaces are implemented as web applications and communicate to the main server through WebSocket connections. For consistency and ease of development, WebSocket messages continue to use the same format as the publisher-subscriber event bus of the main server. The tablet which renders the GUI is connected to the main computer as an external monitor. To align the projected AR view with the real world, we manually calibrate homography transforms for two planes a known distance from each other. From these eight calibration points, the system can calculate a homography for any known plane within the volume.

5.4.5 Enabling an augmented drill, software walkthrough

To concretize the discussion of the software stack, we outline how the drill interaction is coordinated (Figure 5.10). When the tool is selected through the GUI, front-end JavaScript sends a WebSocket message to the main server, which is forwarded to the event bus. This message is caught by the finite state machine (FSM) object. First, the FSM object checks to see that this transition can occur safely. Subsequently, it emits a series of messages for the end-effector electronics, robot, and motion controller objects in order to turn on the force-torque sensor, change the robot's coordinate system for the new tool, and start the motion controller object, respectively. Before allowing motions to occur, the FSM object pauses for three seconds to tare the force torque sensor. The motion manager object is then configured to listen to force-torque messages, calculate the desired motion, and relay the new position and orientation to the robot. Inverse kinematics is handled by the robot manufacturer's controller.

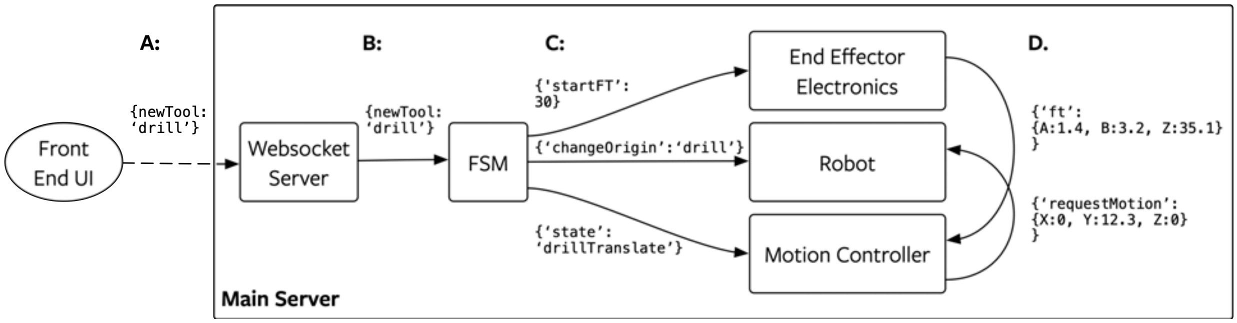


Figure 5.10: A detailed view into the server which coordinates the interaction: Dotted lines represent communication over WebSocket, solid lines represent communication over the event bus. (A) When a tool is selected on the GUI, an event is sent through the WebSocket connection, and is (B) forwarded onto the event bus. (C) After the FSM object confirms that the system can enter the 'drillTranslate' state, it emits a series of events to ready the system for motion. (D) The motion controller object configures itself to listen for force-torque events, and emit the corresponding motions.

5.5 Interviews with expert fabricators

5.5.1 Procedure

To probe reactions to our system, we conducted remote semi-structured interviews with four expert fabricators[3]. Each interview lasted between 60 and 90 minutes. In the first 30 minutes, we interviewed the participants about their experiences with fabrication tools, as well as the challenges that they encounter in their practice. Next, we showed the participants videos of the proposed interactions. This included an introductory video, followed by a short video for each tool: the probe, drill, ruler, and camera. In the remaining time, we discussed their reactions to the different tools, their perceptions of the overall concept, as well as how they envision the system being used in the future.

5.5.2 Participants

We recruited individuals with significant experience across CNC and/or manual fabrication tools. The novelty of our system is primarily within the interactions and workflow. Expert fabricators are often aware of many workflows for accomplishing similar tasks, as well as the specific challenges of each workflow. As such, they are uniquely suited to reflect on how the interactions we developed fit within the landscape of fabrication technologies. Within these constraints, we wanted participants to span a diverse range of disciplines and perspectives for fabrication:

---

[3] Due to the global COVID-19 pandemic during 2020 and 2021, all user studies were conducted remotely.

P1 is a mechanical engineer with ten years of fabrication experience. He has worked as an engineer in a variety of contexts, from underwater vehicles in research and hardware startups, to propulsion and power systems for aerospace vehicles. **P2** is the director of fabrication services within a university architecture program. P2 has over 12 years of professional experience with fabrication, and draws upon a rich background in sculptural and installation art, as well as building and construction in the trades. **P3** has worked as a professional luthier[4] for 18 years and creates instruments primarily with traditional hand tools and techniques. He has explored CNC machines for creating forms and tools, as well as for facilitating repetitive tasks such as shaping the rough geometry of wood. **P4** is the owner of a contract manufacturing shop, which has been in operation for 27 years. P4's shop is equipped with a diverse range of manual and digital fabrication tools, primarily catered to metal-working.

All participants were familiar with both CNC and manual tools. P2 described a philosophy for navigating between different workflows that was echoed by many of the participants: *"For me, it's never an either-or conversation. It's: they're all tools. One is not better than the other, it depends on the application. I don't see it as a competitive thing like manual or digital, to me there's no hierarchy there."* On this topic, the luthier expressed a unique but related perspective. For him, there is no substitute for the trained hand when it comes to finalizing the shape of a violin, because a CNC machine creates the wrong kind of perfect: *"if you make those thicknesses very perfect mathematically, geometrically, the instrument doesn't sound that good."* To create instruments with superior timbre, P3 requires tools that allow him to adapt the design to the needs of the material at fabrication time.

5.5.3 Results

Participants were enthusiastic about the capabilities supported by *aDroid*. Below, we discuss the various ways that participants saw value in our proposed system, as well as new potential applications that the participants suggested.

Overall approach. All participants recognized the value of our high level approach — balancing the directness of manual control with the precision of computer control. P2 expressed that the system is not *"exactly right in the middle or a compromise, but in some ways pulls from the best things from both areas [manual fabrication and digital fabrication]."* The luthier, P3, being the most hand-tool centric participant commented, *"The most important thing is that you're bridging the gap between the hand and the machine. Because that's always been a separate thing, the CNC is so far removed from the hand and the feedback that you can get."*

Physical Jigs/Virtual Jigs. P1-3 all described instances in which they created custom jigs or fixtures to aid in physical fabrication or assembly tasks. When speaking to aspiring professional woodworkers, P2 always emphasizes that *"woodworking is all about problem solving and having an understanding of how to make jigs and fixtures, not just what's out*

---

[4] One who builds stringed instruments, such as violins and cellos.

there but being creative. Making your own jigs, making your own fixtures that are appropriate to your project or need." On a similar note, P3 described how the physical template and mold for his violin designs are the *"seed of the instrument, and then [the violin] just grows around it"*. The value of using using *virtual* jigs and fixtures quickly resonated with P2 and P3 in particular, who work the most with woodworking tools. P3: *"usually in the workshops, the most time that you spend is setting up machinery, and so when you set it up, you want to make sure that you're making many parts at the same time so you're not wasting the setup time. And so having something flexible that you don't have to set up, it's buttons on a computer instead of a physical screw that you have to turn, that would make everything a lot more fluid."*

Tools and Interactions Participants were generally excited by the specific interactions we developed. After seeing the video, P3 exclaimed, *"That's fantastic! So it's like a snap tool, but in real life!"* P1 and P2 also noted specific examples in which the drill tool would have come in handy, for example, drilling holes for trees or streetlamps when building architectural models (P2), or quickly and accurately fabricating an instrument front panel (P1). P1 also envisioned how the probe and drill tool can be used in concert to locate and finalize the hole diameters of cast components, and P2 was excited by the possibility of faster setup time for cutting more complex geometries with the track saw.

Accessibility and Empowerment. As an educator, P2 expressed the most excitement about the possibility for more accessible hand tools. *"This allows anybody shape, size, age, whatever the ability to do these things on their own. For me, that is the most attractive sell... Regardless of precision, quality, control, and all that stuff. For me it's the matter of empowering users."* When the user and robot work with the drill, the robot carries the weight of the tool and braces against the torque generated during cutting — these elements have the potential to bring the satisfaction of working with hand tools to a broader audience.

Limitations of the Approach. P2 brought up one limitation of the track saw. Because *aDroid* only holds the intermediate track rather than the saw itself, if the user is not holding the saw *"tight on the fence, [it] introduces problems that could cause quality issues but also safety concerns"* that are similar to existing manual track saws. We separated the track and saw because the track can potentially be paired with many other tools, such as a router for cutting grooves of precise depth and width, or a pencil for layout. However, because the system does not mediate the position of the tool directly, the quality of the final outcome is more dependent on the user's own skill level.

The examples we showed were focused on woodworking, whereas P4's practice is primarily in metalworking. As a result, P4 was less excited by the specific tools that we showed, but more enthused by how the interactions around force feedback could apply to existing tools he has in his shop. For example, equipping his waterjet with force feedback on the nozzle so that it knows if it has snagged on the workpiece, or being able to manually control a CNC milling machine remotely with direct force feedback. The style of interactions we present may be less appropriate for fabricating metal.

User proposed scenarios. Participants suggested a diverse range of different applications and scenarios in which this type of system can be applied, such as painting (P1), carving with a router (P3), secondary operations (e.g. reaming, tapping, installing threaded inserts, etc) (P1, P4), visualizing how the components of a design can be oriented onto the stock material with AR (P2, P4), and more. Many of these applications are natural extensions of our system; we highlight one envisioned scenario which conceptually deviated the most from the interactions which we presented: P3 suggested how this kind of system can be used to help sharpen his gouges (chisels with concave blades used for carving). He envisions the robot as not only the jig which helps to maintain the same bevel angle across the concave, but also the robotic assistant who finishes sharpening autonomously after he shows it the appropriate way to move across the sharpening stone. P3: *"if you're making a flexible assistant for the shop, you know, there's no limit to what you could be doing with it."* *aDroid* is driven by an actuated robot, and we have the flexibility to explore interactions which feel entirely manual, entirely automatic, or something in between.

5.6 Discussion

We emphasize that this technique is not meant to replace existing CAD software that can represent highly complex geometries, or CAM software that can simulate and optimize toolpaths for metrics like speed or material usage. Instead, *aDroid* seeks to create a lightweight way to engage with a set of these capabilities, without necessarily engaging with the entirety of existing workflows.

5.6.1 Towards more comprehensive interactions

The interactions demonstrated with each tool can be mixed and matched. For example, being able to see the dimension between operations would be helpful for the drill as well as the track saw. Similarly, snap-to-grid would be helpful for many tools, not just the drill. GUI interaction details that would allow the user to input different drill bit diameters, saw blade thicknesses, or adjust the virtual grid width have yet to be implemented. In the current version of the system, engineering efforts have been focused towards demonstrating the core research contribution—using a robot to programmably constrain user motions.

5.6.2 Generalizing *aDroid* beyond its current instantiation

The concepts motivating *aDroid* are not bound to the particular details of its current implementation. For example, many other robots are compatible with the interactions enabled by *aDroid*. Two related characteristics to consider are payload (how heavy of a tool can the robot hold) and stiffness (is the robot suitable for tasks such as CNC machining, which generate additional forces that the robot must resist). However, smaller robots with lower allowable payload can still be used to augment tools. A smaller robot like the Universal Robotics UR3, which has a 3kg maximum payload, may not have the strength to hold a drill, but can hold a drill bushing that can offer similar interactions such as snap-to-grid, or rapidly setting angles.

In addition, different force-torque sensing configurations can be explored. Most force-torque sensors calculate forces by measuring the deflection of rigid springs. Though these deflections are minimal, small angular displacements are amplified by the distance between the sensor and the tool's tip. We were concerned about these small deflections when designing the saw tool. Specifically, one hazard we wanted to avoid altogether was kickback—if a spinning saw blade becomes misaligned with the cut, the blade can push (or "kick") the saw out of the material. This was an additional reason that the user guides the track, rather than the intermediate tool, with the current version of *aDroid*.

However, force-torque sensing can be placed in different locations along the structural loop between the robot and the user. In the current version of the system, the user's desired motions are sensed at the interface between the robot and the tool. Alternatively, sensing can be placed between the tool and the user. In this configuration, the tool will be rigidly coupled to the robot, and whichever force-torque sensor is chosen no longer needs structurally support the tool. However, more extensive modifications on the tools themselves may be required, and none of the forces generated by the tool can be measured.

In terms of safety, future systems can also incorporated more sophisticated precautions than a foot pedal. For example, additional force and touch sensing can be incorporated throughout the robot, rather purely on the wrist. Other forms of sensing, such as depth cameras, can also be leveraged.

5.6.3 Cost

The Kuka robot used in this project costs approximately $30,000 USD. This places it in the high end of tools available within makerspaces, such as CNC mills and waterjet cutters. In the near future, we envision how similar shared usage models can be applied to these robots. Robots for domestic tasks offer a hopeful parallel in terms of decreasing cost: a PR2 robot cost $200,000 USD in 2011, while the recently released Hello Robot is priced at around $20,000 USD. Though current domestic robots are focused on automating tasks such as tidying and laundry, we are looking towards a future in which this technology could be so ubiquitous, that home garages are equipped with collaborative robots that assist with rapid prototyping, home repair, or other DIY tasks. Lower performance robots may still be used for similar interactions

5.7 Future work

Many of the tools and applications suggested by our interviewees are fruitful avenues for future work. In addition, we want to further develop interactions that allow users to fluidly define custom virtual constraints in order to adapt the robot for new tasks. More broadly, we are interested in exploring how this kind of system can directly communicate embodied tacit knowledge. Tacit knowledge [93], such as the ability to ride a bicycle, is knowledge that is easily known but difficult to communicate. Consider the tasks of learning to throw a frisbee or refining a tennis swing — what if these skills can be shown directly through a software defined 'jigs' that can directly convey how the motion should feel through force feedback, rather than implicitly transmitted through verbal or visual instructions? Though this research is focused heavily on direct and hands-on interactions, it does not preclude opportunities for autonomous robotic control. We are particularly interested in the exploring the concept of reciprocal agency — how might users fluidly navigate between direct control, computer control, and the ambiguous spectrum in between? A simple example is "copy & paste". Because all motions are mediated by through the robot, this system is uniquely positioned to record and playback repeated tasks, such as the sharpening scenario posed by the luthier.

5.8 Conclusion

aDroid supports the process of hands-on fabrication through virtual constraints rendered through a robotic arm. Through these constraints, users can borrow from the accuracy and precision of the robot for hands-on fabrication tasks. In this chapter, I described the design, implementation, and usage of such a system. Revisiting our drill example, the robotically augmented drill is much more than a computationally generated drill press. It is imbued with snap-to-grid for easily placing accurately spaced cuts; it can remember where holes have been drilled and how the workpiece has been modified; its settings can be instantly saved, restored, and shared. Though initially inspired by the unique affordances of physical jigs and fixtures, *aDroid* leveraged the concept of a digital jig further as a means to incorporate digital editing affordances directly within physical fabrication. The computer is tasked now not only with aiding digitally, but helping orchestrate the design and fabrication process in physical space, in real time, and in-situ.

Chapter 6

Conclusion

In order to turn an idea of an object into a fabricated object, a user must navigate a plethora of design considerations about the overall structure of that object, its geometric details, the materials it will it be made from, the appropriate process and method for fabrication, as well as the many ways in which these considerations intersect. Within this open ended, yet highly coupled, design space are countless opportunities to exercise human judgement, skill, and creativity.

However, conventional workflows for digital fabrication explicitly exclude these opportunities from the operation of the fabrication tool—all design decisions must be formalized through software before any fabrication can commence. In response to the limitations imposed by this approach, we began with the following motivating question: *how can digital fabrication tools engage opportunities for human judgement, skill, and creativity during the fabrication process?* This dissertation presented one approach: allow users to determine what to make, and how to make it, while operating the fabrication tool. When aided by techniques and interactions that support the process of hands-on fabrication, users can access the benefits of digital fabrication while departing from conventional workflows.

With *MatchSticks*, I considered how existing CAD/CAM software can be distilled and incorporated directly at the machine itself. This entire system, from software interactions to the physical machine, was tailored to the particular needs of fabricating wooden joinery. The resulting interaction can be likened to how a user would interact with an electric miter saw. The user first sets up the machine for a desired geometry, then uses that machine to power through the cut. With *MatchSticks*, the user leverages not only electrical power, but computation to create more complex geometries.

This system supports some level of design directly, in terms of allowing its users to select the appropriate joint for particular object. More importantly however, *MatchSticks* allows for more involved design considerations, such as those about the overall structure of an assembly, to be navigated outside of conventional CAD software. This is not to say that CAD and *MatchSticks* are incompatible; rather, CAD may be unnecessary for managing certain designs because *MatchSticks* abstracts away the component level complexity of joinery, and the overall assembly level complexity can be relatively simple for many wooden objects.

With *Turn-by-Wire*, I considered the complement of control—feedback—as well as interactions that allow users to create open-ended geometries. While users work with the haptic handwheels, they are not only controlling the position of the tool, but receiving feedback from the machine. With "by-wire" control, haptic feedback need not be limited to literally communicating the forces experienced by the tool. Combined with new interactions for remapping control, *Turn-by-Wire* enabled a set of interactions that are more often associated with digital editing, rather than physical editing.

When users work with *Turn-by-Wire*, they are navigating considerations around how an object should be made. Like a manual tool, users interact with *Turn-by-Wire* continuously by hand. As a result, the user can instantaneously adapt and refine the way an object is made—the overall strategy, as well as the speeds, feeds, and depth of cut—in response to the tool and the material. In contrast, refining these types of decisions with CAM requires the user to alternate between the software which generates the toolpath and the physical machine which executes the toolpath. In addition, because all of a user's actions can be recorded through the system, the process of fabricating an object simultaneously generates a digital representation of that object, as well instructions for how to make that object. In this sense, *Turn-by-Wire* shares overlapping functionality as CAD and CAM software.

Lastly, I explored how the ideas that emerged from *Turn-by-Wire* could be generalized in an extensible way. Rather than augmenting tools individually, I considered how *aDroid* could couple "by-wire" control directly to the user. Specifically, I explored how the framework of virtual jigs and fixtures can support users in open ended fabrication. For breadth, I demonstrate four tools that could be used with the system, each of which showcase a specific interaction related to capabilities of digital editing. Like *Turn-by-Wire*, *aDroid* allows users to design objects and design how to make objects while directly working with fabrication tools. Whereas *Turn-by-Wire* focused on one tool in particular, *aDroid* sought to extend this concept to the fabrication environment as a whole.

Collectively, these three projects demonstrate that the benefits associated with digital fabrication can be achieved through workflows that deviate significantly from conventional ones. Hands-on control was not leveraged solely for the sake of tangibility or directness, but as an alternative modality that, when mediated through computation, allows users to fabricate complex objects accurately.

6.1 Limitations

The approaches presented in these projects present exciting new ways of engaging with digital fabrication tools. However, none of these systems should be interpreted as replacements for existing tools, interactions, or workflows. Instead, each fabrication system trades one set of advantages for another.

6.1.1 Negotiating risk

One of these trade-offs is the types of risk that different fabrication methods entail. Working in real time with physical tools and materials presents a unique kind of risk—there is no undo button in real life. However, more commonplace digital fabrication workflows are not without their own kinds of risk. When fabricating objects with a CNC router, there is the risk of the machine crashing during operation because of an incorrectly parameterized toolpath. When printing a design with a 3D printer, there is the risk of waiting for hours for a print to finish, only to find that some key dimensions or tolerances where incorrect designed or incorrectly printed. And when the process of fabrication is abstracted away from the process of design, there is a risk of painstakingly designing an object, only to find out that it is impossible or impractical to physically fabricate. More disastrously, an architecture firm may design a facade that appears beautiful in simulation, only to discover that it is hideous when fabricated [125].

6.1.2 Negotiating control

Interactions involving continuous embodied control were leveraged significantly in *aDroid* and *Turn-by-Wire*. In each of these projects, the user never moved the tool directly, instead, their motions were remapped through some kind of controller. In some ways, this style of control has the potential to make physical fabrication more accessible to users with different degrees of mobility, range of motion, or physical strength. A lathe can be operated sitting down, and a hand drill's weight and torque can be mechanically braced. However, for designers with more limited range of motion, systems which seek to automate more and more of physical fabrication may be more appropriate and empowering.

6.1.3 Negotiation complexity

To create the electromechanical systems presented in this dissertation, I leveraged existing CAD/CAM workflows extensively in order to manage the complexity within the designs, refine their details, and simulate their performance before fabrication. Most likely, I would not have been able to "improvise" a force-torque sensor (Appendix A) with the types of tools presented in this dissertation after making a few sketches with pen and paper. Tools like *Turn-by-Wire* and *aDroid* address component level complexity, but not assembly level complexity. *MatchSticks* addresses assembly level complexity, but only as a byproduct of significantly simplifying the domain. As outlined in Figure 2.5, these types of tools are not incompatible with CAD. For example, I designed *aDroid*'s touchprobe in CAD, and fabricated one of its subcomponents, the insulated bushings, using *Turn-by-Wire*.

6.2 Automation and augmentation in tension

Within society at large, technological unemployment has been a persistent concern. Technologies which clearly seek to automate human labor—such as self driving cars—directly aggravate these concerns. Simultaneously, technologies characterized as augmenting rather than replacing humans should not necessarily allay these concerns. The overlapping intent of automation and augmentation are highlighted by early systems like *record-playback*, in which a machinist could define how components should be automatically fabricated by manually machining the first component. One interpretation of *record-playback* is that it valued and sought to augment the machinists' manual skill. Simultaneously, its creators cite that one of the benefits of this system is that "operators' errors are eliminated" [84], suggesting an underlying desire to reduce reliance on skilled labor that was not so different from the motives driving numeric control.

The projects in this dissertation are deeply motivated by a desire to augment the process of hands-on fabrication. However, each system facilitated fabrication by requiring less time, energy, or expertise from the user. While none of these fabrication tools can accomplish anything without a user, varying degrees of automation accompany each system. In some cases, automation is explicitly leveraged as a means for augmentation: *MatchSticks* fabricated joinery through computer control rather than manual skill, and *Turn-by-Wire* alleviated repetitive movements by revisiting interactions similar to *record-playback*. However, these projects sought to carefully balance how humans and computational tools participate in the fabrication process: users are empowered to exercise creativity and judgment while being supported by the flexibility, accuracy, and strength of computational tools.

6.3 Who might these "products" be for?

The primary motivation for each of these projects was to articulate concepts and directions that challenge existing digital fabrication workflows. Though the ideas embedded within each project are relevant for digital fabrication in general, these ideas were instantiated through specific systems that made specific types of things. As such, it is possible to interpret each of these systems as "products," which naturally leads to the question of who these "products" are for. Here, we reflect on potential groups who are likely to find value in these particular instantiations.

The *MatchSticks* and *aDroid* systems are woodworking focused. Both could potentially find homes in makerspaces. Though industrial robots are expensive, their cost is still within the range of tools like CNC routers, which are commonly found in makerspaces. By the same token, it is unlikely that many hobbyists will have access to this kind of system in the near future. *aDroid* could also be applicable to commercial woodworking. Existing industrial table saws sometimes have computer actuated fences and trunnions to ensure accurate setups. Like these more commercially oriented woodworking tools, *aDroid* could be used to ensure precision for a broader set of tools. It is also possible for *MatchSticks* to be utilized in professional shops alongside larger CNC routers, where smaller parts could be machined using *MatchSticks*, rather than tying up the larger CNC machines. As a smaller and cheaper machine, *MatchSticks* may find a home in a hobbyists garage. A commercial product similar to *MatchSticks* is the Pantorouter [130], which shares similar functionality and layout but is controlled by physical templates rather than computation.

The *Turn-by-Wire* system may be valuable for low-volume machine shops. During the *aDroid* interviews, the owner of the contract manufacturing shop described how he wanted to explore precisely the same idea that motivated *Turn-by-Wire*—using force feedback handwheels so that he could rapidly dial in the feeds and speeds for the countless combination of tools and materials that he may have to work with. *Turn-by-Wire* might also be used as a teaching tool for machinists, since the force feedback can signal best practices for working with certain tools. Again, because of the desktop form factor, hobbyists may find an interest. Many hobbyist desktop manual and CNC machine tools are currently available; the Sherline brand is one example. The control component of *Turn-by-Wire* may also be valuable as a retrofit kit for hobbyists to improve their existing manual or CNC machine tools.

6.4 Future work

Specifically, this dissertation focused on digital fabrication. Thematically, my work considers how technology might engage and extend human capabilities, illuminate tacit knowledge, support creative practices, and understand user workflows. In future research, I will continue to build and evaluate interactive systems to broadly explore these themes. Below, we discuss future work, from concrete proposals to speculative envisionments. Future directions relating to *MatchSticks*, *Turn-by-Wire*, and *aDroid* specifically can be found in their respective chapters.

6.4.1 From maximal to minimal interventions

aDroid introduced capabilities more often associated with digital editing to the process of physical fabrication. It is the most speculative of the three projects, and functions the most as a case study of interactions, rather than as a road-map for commercialization. However, ideas embodied by *aDroid* can be distilled and integrated within fabrication tools in more subtle ways.

Consider the fence attachment for a router, which allows a user to cut at a consistent distance away from the edge of a workpiece. A simple one degree of freedom sensor or actuator for setting this distance can be used to achieve automatic accuracy—neither a robot nor complex computer control is required. As with other computationally augmented manual tools, a similar spectrum of interactions are possible here. The system could simply display the distance that the fence is currently located. Or it can be actuated, where the user inputs a desired distance, and the system positions the fence accordingly. As always, the interaction could blend the above modalities—the user can move the fence directly, but sensing and actuation render physical detentes for locations of interest. Another example is the circle cutting jig, which functions like a compass—the jig is anchored at the center of the circle, and a cutting tool can be swept along the circumference. In the spirit of [78], if the radius of the circle jig can be actuated, the jig could be used just as easily to cut ellipses, squares, text, or more complex geometries.

This proposed work shares a similar motivation with *aDroid*, which is to facilitate how users set up and use manual tools. Instead of addressing the general problem with an industrial robot, a complementary approach is to consider how computation can be selectively applied with restraint.

6.4.2 Abstracting continuous control

With *MatchSticks*, users specified a desired geometry to the tool, but unlike with *Turn-by-Wire* or *aDroid*, they were never directly in control of the fabrication process. During the user study, participants more familiar with hand tools simultaneously felt excited about making joinery quickly and accurately, yet estranged from the actual process of fabrication. This tradeoff is not inevitable; interaction strategies from the other two works can be readily adapted. By integrating handwheel control, users could move the three axis of the machine independently; with force sensing on the electric router itself, users could move the cutting tool directly by hand.

Alternatively, we can begin to explore more abstract, but no less direct, strategies for continuous control. Touch screens, 3D mice, or even video game controllers can be used to continuously control the position of a fabrication tool—the machine used in *MatchSticks* can be utilized as a system for realizing and testing these concepts. To enable users to fabricate accurately with any of these controllers, the output motions can be constrained by software, similar to the virtual tools of *Turn-by-Wire* or the constraints in *aDroid*. After the user demonstrates how to cut a particular geometry, or demonstrate what sensible speeds and feeds are for a particular piece of lumber, subsequent operations can potentially be completed automatically. In addition, with both continuous and conversational interactions integrated within one system, questions around when, why, or which users choose one interaction over the other can be precisely probed.

6.4.3 Digital fabrication for a luthier

During the *aDroid* interviews, the luthier described how existing CNC workflows are uniquely ill suited for his practice. First, being able to fabricate geometrically perfect components is neither necessary nor desired. He spoke about some violins made at the turn of the 20th century that produce excellent sound; however, the insides of these instruments look like they were "carved with squirrel teeth." The natural variations of manual carving is not a liability to be automated, but an opportunity that can be practically leveraged to produce instruments with nuanced timbre.

When working with precious materials, like the wood used in violin making, the potential risks of a CNC programming error may outweigh the benefits of the overall workflow. The luthier showed one piece of wood that he has been saving and drying, the only piece that he has remaining from a tree that was felled decades ago. CNC tools may be able to create geometries faster and with less physical effort, but these benefits carry less weight for an irreplaceable material that can be ruined instantly by an errant line of G-code. So far, the luthier has only used CNC machine to fabricate jigs and forms from plywood, rather than the instruments themselves.

Lastly, the CNC machines available to him are completely material unaware. *aDroid* and *Turn-by-Wire* touches on these topics, but the luthier re-emphasized the role that direct physical feedback can play during the fabrication process. When carving a violin, the luthier adapts both his technique and the desired geometry depending on the grain and density of the wood, which is revealed only through and during the process of fabrication.

One way to address these challenges is to collaborate with the luthier to design and create a new digital fabrication system. This resulting artifact could build upon a general purpose robot like *aDroid*, or perhaps consist of a bespoke machine. Methodologically, this presents a unique opportunity to work directly with a single "target user", who can guide the focus and scope of the system's development. Of the nearly endless array of jigs and fixtures for making stringed instruments, which can or should be augmented through computation? When working with precious materials, how might computational fabrication tools be designed to foster trust and facilitate transparency? Which aspects of a deeply hands on process should be automated—where precisely is the line between meditative and monotonous, and how and when might this boundary shift? By focusing on a domain where the challenges of CNC workflows are uniquely amplified, this proposed work seeks to uncover new interactions and capabilities that have the potential to benefit all users who work with digital fabrication tools.

6.4.4 Reciprocal agency

Beyond simply copy-paste or record-playback, how can users and fabrication systems work collaboratively throughout complex fabrication tasks? How might a the system autocomplete a users actions, or how might a user mediate computer guided operation? In these scenarios, agency—and authorship of the final artifact—reciprocates between the user and the tool. While certain fabrication tasks are best suited for either manual or computer controlled fabrication, this binary choice lacks the flexibility to adapt to rapid prototyping workflows. How might users or machines initiate transitions into this liminal space during fabrication? What modalities might be used to communicate between users and machines: sounds, subtle motions and gestures, or haptic cues? How might our tools exhibit personality, gesture intention, or communicate uncertainty, to collaborate with us in more personal and expressive ways?

6.4.5 The gardener and the architect

Both *MatchSticks* and *Turn-by-Wire* explored how the tool itself can scaffold novices. In *MatchSticks*, on-screen tutorials guided users through the process of building parametric woodworking primitives; in *Turn-by-Wire*, force feedback communicated what techniques where appropriate for a particular tool. Moving forward, I am interested in further investigating how interactive tutorials can support open-ended exploration as an alternative to step-by-step scaffolding. If users rely on incremental instructions as the primary way of learning a new skill, how might they be empowered to confidently solve novel problems? What are the trade-offs in long term learning outcomes between structured and open-ended instructional styles? I propose creating and evaluating tutorial environments that emphasize teaching the process of using a tool, rather than instructing users to create a particular product. By studying and understanding different styles of guidance, we can accelerate how users gain new skills through guidance that is adaptable not only to skill and context, but higher level learning goals.

6.5 Summary

All three approaches presented in this dissertation acknowledge and celebrate the process of hands-on fabrication as a site of creative exploration and problem solving. In *MatchSticks*, I explored how existing Computer Aided Design and Manufacturing workflows can be distilled and incorporated within the fabrication machine itself. With *Turn-by-Wire*, I explored how users can continuously control a lathe by hand, while gaining capabilities associated with digital editing. Lastly with *aDroid*, I explored how the the ideas with *Turn-by-Wire* can be extensible to many tools, by augmenting the user and environment directly, rather than each tool individually. None of these proposed systems are intended to replace any existing tools, interactions, or workflows. Instead, this work simply seeks to expand the “toolbox” of techniques, so that it can be more adaptable and appropriate for different users, practices, materials, designs, and contexts.

Appendix A

Creating a low cost, high resolution force–torque sensor

Force and torque sensing are essential to enabling the interactions explored in *aDroid*. However, commercial force-torque sensors can cost upwards of $5000 USD, which is nearly 20% the cost of the entire robot. Not only are the cost of robots themselves a barrier to entry for researchers, so are their accompanying accessories. I endeavored to build a sensor that was high resolution, easy to manufacture, and low cost. This prototype sensor was fabricated at an academic makerspace for under $100 USD, and measures two axis of torque (X and Y) and one axis of force (Z). I characterized the z-axis performance (Figure A.1 bottom-right) and measured a signal noise (standard deviation of the signal over 100 samples) of 0.1 Newton. This resolution is comparable to existing commercial force-torque sensors today. I developed this sensor only as far as to support the interactions described in *aDroid*, and additional refinement and characterizations are possible. However, this sensor architecture can potentially become a viable starting point for researchers who want to explore the unique possibilities afforded by force based interactions.

Like most force torque sensors, this sensor measures forces and torques via displacements. The primary mechanism of the sensor is a plate (Figure A.1 left) which consists of an inner and outer ring connected by thin members designed to elastically deform under load. These thin members are typically referred to as flexures. By measuring the displacement of inner ring with respect to the outer ring, the applied forces can be calculated given the stiffness of the mechanism (i.e. $F = kx$). In this system, displacements are measured through through the inductance shift of printed circuit board (PCB) coil inductors.

The mechanical structure of the sensor is configured to facilitate manufacturing and design. To avoid the difficulties of machining (fixturing, toolpath generation, etc), this sensor consists of a stacked assembly of waterjet aluminum plates (Figure A.1 center). In addition, this configuration supports modular iteration—if a component needs to change, only that layer of the stack-up needs to be fabricated again. Flexural mechanisms have been widely studied [114, 46], and many can be applied to this design. Ultimately, I settled on a design consisting of radially arranged flexural hinges, whose stiffness with respect to its dimensions

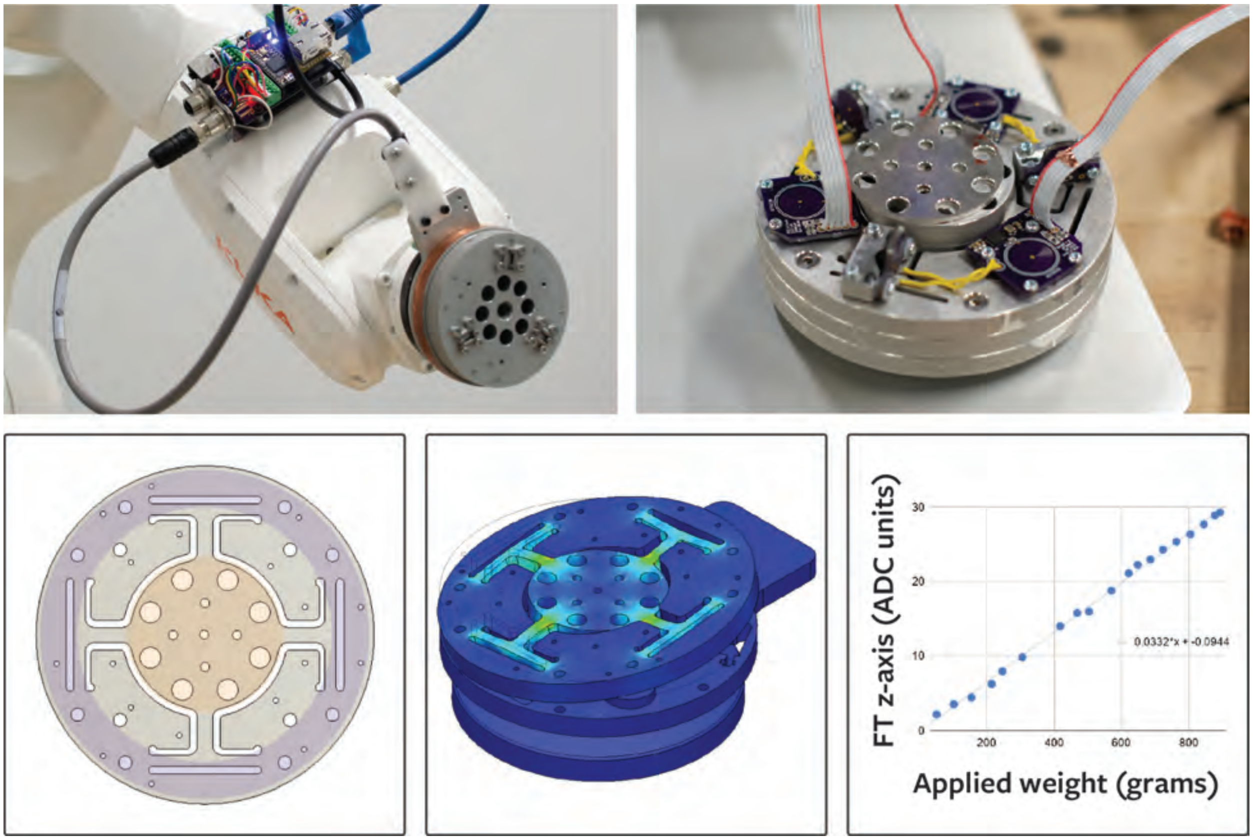


Figure A.1: (A) Force torque sensor mounted to the end of the robot. (B) Inside view of the force torque during assembly. Each of the printed circuit boards contain a spiral inductor for distance sensing. (C) Flexure detail. The center disk (highlighted orange) is attached to the wrist of the robot, and the outer ring (highlighted purple) is attached to various end effectors. These rings are joined by thin T-shaped flexures which are designed to elastically deform. (D) FEA studies of the force torque sensors help ensure that applied forces and resulting deflections are measurable by our sensors and tolerable by the material. Displacements are exaggerated. (E) Plot of z-axis signal response with respect to applied weight.

is easier to understand and characterize. After settling on the overall shape of the flexural plate, FEA studies were conducted to fine tune the dimensions of the flexure to confirm that typical forces would generate deformations that are measurable by the electronics and tolerable by the material (Figure A.1 center).

Resistance, capacitance, and inductance can all be leveraged to measure displacement. Mechanical strain can be measured directly with strain gauges, but inductive and capacitive approaches share the advantage of being contactless. I chose to measure displacements using an off the shelf, 28-bit inductance to digital converter integrated circuit (Texas Instruments

LDC1612, used in metal proximity sensing). The inductance being measured is that of a PCB spiral inductor, whose effective inductance changes as a function of its distance to nearby metals. Three PCB inductors are evenly arranged along the outer perimeter of the flexure plate, giving us three measurements for out of plane deflection at these points. By assuming that the outer rings stay rigid, and that only axial displacements are being measured by each coil, we can calculate the axial displacement (Z) as well as the tilt about the two other orthogonal axis (X and Y).

I conducted an initial experiment to measure the sensor's resolution in the z-axis only. A 0.1 gram resolution digital scale was placed on top of the sensor, and an empty plastic cup was placed on top of that. I then tare both the digital scale and the force torque sensor. I poured water into the cup, and took measurements at approximately 50 gram intervals. The plot of measured force vs the force torque sensors z axis output can be seen on the right panel of Figure A.1. Under a static load, the standard deviation of the signal over 100 samples was measured to be 0.1 Newtons in a home office environment.

Because of its high sensitivity, this sensor shares similar shortcomings as commercial force-torque sensors. For example, it is also a temperature sensor as a result of thermal expansion and contraction. More importantly, it is sensitive to strong nearby magnetic fields — a magnetized screwdriver held directly next to the sensor will noticeably affect the sensor output. However, in the tests with the robot, it is not strongly affected by the robot's or drill's motors. This may be a result of the fact that the magnetic flux within motors are relatively well contained compared to the open field magnet of the screwdriver. Mu-metal, a high permeability nickel alloy used for magnetic shielding, can potentially be explored in future iterations.

Bibliography

[1] Harshit Agrawal et al. “Protopiper: Physically sketching room-sized objects at actual scale”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 427–436.

[2] Lea Albaugh et al. “Enabling Personal Computational Handweaving with a Low-Cost Jacquard Loom”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–10.

[3] Chris Anderson. “In the Next Industrial Revolution, Atoms Are the New Bits”. In: *Wired* (2010). ISSN: 1059-1028. (Visited on 06/16/2021).

[4] David Anderson et al. “Tangible interaction + graphical interpretation: a new approach to 3D modeling”. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 393–402.

[5] Erik Andreassen et al. “Efficient topology optimization in MATLAB using 88 lines of code”. In: *Structural and Multidisciplinary Optimization* 43.1 (2011), pp. 1–16.

[6] Ryan Arisandi et al. “Enjoying virtual handcrafting with tooldevice”. In: *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012, pp. 17–18.

[7] Moritz Bächer et al. “Spin-it: Optimizing moment of inertia for spinnable objects”. In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), pp. 1–10.

[8] Akash Badshah et al. “Gyrotab: A handheld device that provides reactive torque feedback”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 3153–3156.

[9] *Baubot—Mobile Construction Robots*. URL: `https://www.baubot.com/` (visited on 06/24/2021).

[10] Patrick Baudisch and Stefanie Mueller. “Personal Fabrication”. In: *Foundations and Trends in Human-Computer Interaction* 10.3-4 (2017), pp. 165–293. ISSN: 1551-3955.

[11] Patrick Baudisch et al. “Kyub: A 3d editor for modeling sturdy laser-cut objects”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.

[12] Hrvoje Benko et al. “Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology.* 2016, pp. 717–728.

[13] Brian T Bethea et al. “Application of haptic feedback to robotic surgery”. In: *Journal of Laparoendoscopic & Advanced Surgical Techniques* 14.3 (2004), pp. 191–195.

[14] *Box.* 2013. URL: `https://youtu.be/lX6JcybgDFo` (visited on 06/24/2021).

[15] Garrett W. Brown. “Equipment for use with hand held motion picture cameras”. US4017168A. Apr. 1977.

[16] Giulio Brugnaro and Sean Hanna. “Adaptive robotic training methods for subtractive manufacturing”. In: *Proceedings of the 37th annual conference of the association for computer aided design in architecture (ACADIA).* Acadia Publishing Company. 2017, pp. 164–169.

[17] *Canvas—The Future of Construction.* URL: `https://www.canvas.build/` (visited on 06/24/2021).

[18] Minsuk Chang et al. “Workflow graphs: A computational model of collective task strategies for 3D design software”. In: *Graphics Interface 2020, GI 2020.* Canadian Information Processing Society. 2020.

[19] Amy Cheatle and Steven J Jackson. “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.* 2015, pp. 958–968.

[20] Inrak Choi et al. “Grabity: A wearable haptic interface for simulating weight and grasping in virtual reality”. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology.* 2017, pp. 119–130.

[21] *CNC machine tool manufacturing history.* URL: `https://www.hurco.com/en-us/about-hurco` (visited on 06/24/2021).

[22] *Conversational Programming in Tormach’s PathPilot.* Oct. 2016. URL: `https://www.youtube.com/watch?v=gd7IzOGtFKY` (visited on 06/24/2021).

[23] Chengkai Dai et al. “Support-free volume printing by multi-axis motion”. In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), pp. 1–14.

[24] Pradeep Devadass et al. “Robotic fabrication of non-standard material”. In: *Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)* (2016), pp. 206–213.

[25] Laura Devendorf and Kimiko Ryokai. “Being the machine: Reconfiguring agency and control in hybrid fabrication”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems.* 2015, pp. 2477–2486.

[26] S Devine, K Rafferty, and D Ferguson. "HapticVive—a point contact encounter haptic solution with the HTC VIVE and Baxter robot". In: *The 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2016 in co-operation with EUROGRAPHICS* (2017), pp. 17–23.

[27] Fotios Dimeas et al. "Manipulator performance constraints in cartesian admittance control for human-robot cooperation". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 3049–3054.

[28] Dale Dougherty. "The maker movement". In: *Innovations: Technology, governance, globalization* 7.3 (2012), pp. 11–14.

[29] Volodymyr Dziubak et al. "Maestro: Designing a System for Real-Time Orchestration of 3D Modeling Workshops". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 2018, pp. 287–298.

[30] *EksoZeroG, Zero Gravity Tool Assistance*. URL: `https://eksobionics.com` (visited on 06/24/2021).

[31] *Epur, Portable Digital Milling*. URL: `https://www.epur.io/` (visited on 06/24/2021).

[32] Sean Follmer et al. "CopyCAD: remixing physical objects with copy and paste from the real world". In: *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*. 2010, pp. 381–382.

[33] Frikk H Fossdal et al. "Fabricatable machines: A toolkit for building digital fabrication machines". In: *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 2020, pp. 411–422.

[34] Chi-Wing Fu et al. "Computational interlocking furniture assembly". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–11.

[35] Madeline Gannon. *Quipt—ATONATON*. 2015. URL: `https://atonaton.com/quipt` (visited on 06/24/2021).

[36] Madeline Gannon. *Robo.Op*. 2015. URL: `http://www.madlab.cc/robo-op` (visited on 01/09/2020).

[37] Madeline Gannon, Tovi Grossman, and George Fitzmaurice. "ExoSkin: On-body fabrication". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 5996–6007.

[38] Jose Luis García del Castillo y López. "Machina. NET: A Library for Programming and Real-Time Control of Industrial Robots". In: *Journal of Open Research Software* 7 (2019), p. 27.

[39] Neil Gershenfeld. "How to make almost anything: The digital fabrication revolution". In: *Foreign Aff.* 91 (2012), pp. 43–57.

[40] David J Gingery. *The Metal Lathe*. Vol. 2. David J. Gingery Publishing, LLC, 2014.

[41] Sidhant Gupta et al. "SqueezeBlock: using virtual springs in mobile devices for eyes-free interaction". In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology.* 2010, pp. 101–104.

[42] Gary S Guthart and J Kenneth Salisbury. "The Intuitive$^{\text{TM}}$ telesurgery system: overview and application". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation.* 2000, pp. 618–621.

[43] *Haas Electronic Handwheels.* 2017. URL: `https://www.haascnc.com/productivity/product-options/e-handwheels.html` (visited on 06/24/2021).

[44] Blake Hannaford and Allison M. Okamura. "Haptics". In: *Handbook of Robotics.* Springer, 2008, pp. 719–739.

[45] *Haptic Intelligentsia - Joong Han Lee/Studio Homunculus.* URL: `https://vimeo.com/43664927` (visited on 06/24/2021).

[46] Larry L Howell et al. *Handbook of compliant mechanisms.* Wiley Online Library, 2013.

[47] Yijiang Huang et al. "Framefab: Robotic fabrication of frame shapes". In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), pp. 1–11.

[48] *Introducing a new construction robot. Hilti Jaibot for construction automation of overhead drilling.* 2020. URL: `https://www.youtube.com/watch?v=ytpzb0tNo3w` (visited on 06/24/2021).

[49] Michael D Jones, Kevin Seppi, and Dan R Olsen. "What you sculpt is what you get: Modeling physical interactive devices with clay and 3d printed widgets". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* 2016, pp. 876–886.

[50] *Joyn Machine—Studio Milz.* URL: `https://milz.it/work/joyn-machine` (visited on 06/24/2021).

[51] H. Kazerooni and Jenhwa Guo. "Human Extenders". In: *Journal of Dynamic Systems, Measurement, and Control* 115.2B (June 1993), pp. 281–290.

[52] Jeeeun Kim et al. "Compositional 3D printing: expanding & supporting workflows towards continuous fabrication". In: *Proceedings of the 2nd ACM Symposium on Computational Fabrication.* 2018, pp. 1–10.

[53] John Klein et al. "Additive manufacturing of optically transparent glass". In: *3D Printing and Additive Manufacturing* 2.3 (2015), pp. 92–105.

[54] Robert Kovacs et al. "Haptic PIVOT: On-Demand Handhelds in VR". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 2020, pp. 1046–1059.

[55] Torsten Kröger. "Opening the door to new sensor-based robot applications—The Reflexxes Motion Libraries". In: *2011 IEEE International Conference on Robotics and Automation.* IEEE. 2011, pp. 1–4.

[56] *KUKA ready2_pilot*. 2020. URL: `https://www.youtube.com/watch?v=8yGaMFDuFbE` (visited on 06/24/2021).

[57] Ben Lafreniere and Tovi Grossman. "Blocks-to-cad: A cross-application bridge from minecraft to 3d modeling". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 2018, pp. 637–648.

[58] Benjamin Lafreniere et al. "Crowdsourced fabrication". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 2016, pp. 15–28.

[59] Maria Larsson, Hironori Yoshida, and Takeo Igarashi. "Human-in-the-loop fabrication of 3D surfaces with natural tree branches". In: *Proceedings of the ACM Symposium on Computational Fabrication*. 2019, pp. 1–12.

[60] Maria Larsson et al. "Tsugite: Interactive Design and Fabrication of Wood Joints". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 317–327.

[61] Manfred Lau et al. "Converting 3D furniture models to fabricatable parts and connectors". In: *ACM Transactions on Graphics (TOG)* 30.4 (2011), pp. 1–6.

[62] Nicolas Lauzier et al. "Force/torque sensor, apparatus and method for robot teaching and operation". US9696221B2. July 2017.

[63] Jay Lee et al. "HandSCAPE: a vectorizing tape measure for on-site measuring applications". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2000, pp. 137–144.

[64] Danny Leen, Raf Ramakers, and Kris Luyten. "StrutModeling: a low-fidelity construction kit to iteratively model, test, and adapt 3D objects". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 2017, pp. 471–479.

[65] Danny Leen et al. "JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.

[66] Jingyi Li et al. "Direct and immediate drawing with CNC machines". In: *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*. 2017, pp. 1–2.

[67] *Machines that Make*. URL: `https://mtm.cba.mit.edu/` (visited on 06/18/2021).

[68] Shiran Magrisso, Moran Mizrahi, and Amit Zoran. "Digital joinery for hybrid carpentry". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–11.

[69] James McCrae, Nobuyuki Umetani, and Karan Singh. "FlatFitFab: interactive modeling with planar sections". In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 2014, pp. 13–22.

[70] *MILLPWR G2 Control Systems*. URL: `https://acu-rite.com/millpwr-control-systems.html` (visited on 06/24/2021).

[71] *Mimic for Maya*. 2018. URL: `https://www.mimicformaya.com` (visited on 06/24/2021).

[72] Koichi Morishige and Miharu Nakada. "Development of turning machine operation interface that uses haptic device (application to complicated cutting by special byte)". In: *Procedia CIRP* 57 (2016), pp. 649–652.

[73] Catarina Mota. "The rise of personal fabrication". In: *Proceedings of the 8th ACM Conference on Creativity and Cognition*. 2011, pp. 279–288.

[74] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. "LaserOrigami: laser-cutting 3D objects". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 2585–2592.

[75] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. "Interactive construction: interactive fabrication of functional mechanical devices". In: *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012, pp. 599–606.

[76] Stefanie Mueller et al. "FormFab: continuous interactive fabrication". In: *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 2019, pp. 315–323.

[77] Stefanie Mueller et al. "WirePrint: 3D printed previews for fast prototyping". In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 2014, pp. 273–280.

[78] Ken Nakagaki and Yasuaki Kakehi. "Comp*pass: A compass-based drawing interface". In: *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 2014, pp. 447–450.

[79] Yoshihiko Nakamura, Kousuke Kishi, and Hiro Kawakami. "Heartbeat synchronization for robotic cardiac surgery". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. Vol. 2. IEEE. 2001, pp. 2014–2019.

[80] Vidya Narayanan et al. "Automatic machine knitting of 3D meshes". In: *ACM Transactions on Graphics (TOG)* 37.3 (2018), pp. 1–15.

[81] David F. Noble. *Forces of production: a social history of industrial automation*. New York: Oxford University Press, 1986.

[82] Alex Olwal, Jonny Gustafsson, and Christoffer Lindfors. "Spatial augmented reality on industrial CNC-machines". In: *The Engineering Reality of Virtual Reality 2008*. Vol. 6804. International Society for Optics and Photonics. 2008, p. 680409.

[83] *Part Programming - Intro 2*. URL: `https://youtu.be/LqrdyxL-Qcs` (visited on 06/24/2021).

[84] L. R. Peaslee. "Tape-controlled machines". In: *Electrical Manufacturing* 52 (Nov. 1953), pp. 102–108.

[85] Nadya Peek and Ilan Moyer. "Popfab: A case for portable digital fabrication". In: *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 2017, pp. 325–329.

[86] Nadya Peek et al. "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*. 2017, pp. 3657–3668.

[87] Huaishu Peng, Amit Zoran, and François V Guimbretière. "D-coil: A hands-on approach to digital 3D models design". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 1807–1815.

[88] Huaishu Peng et al. "On-the-fly print: Incremental printing while modelling". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 887–896.

[89] Huaishu Peng et al. "RoMA: Interactive fabrication with augmented reality and a robotic 3D printer". In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–12.

[90] Michael A Peshkin et al. "Cobot architecture". In: *IEEE Transactions on Robotics and Automation* 17.4 (2001), pp. 377–390.

[91] Wendy Plesniak and Ravikanth Pappu. "Coincident display using haptics and holographic video". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1998, pp. 304–311.

[92] Christopher J Ploch et al. "Haptic skin stretch on a steering wheel for displaying preview information in autonomous cars". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 60–65.

[93] Michael Polanyi. *The tacit dimension*. London: Routledge & K. Paul, 1967.

[94] Marshall Prado et al. "Core-less filament winding". In: *Robotic fabrication in architecture, art and design 2014*. Springer, 2014, pp. 275–289.

[95] Zhan Fan Quek et al. "Sensory substitution and augmentation using 3-degree-of-freedom skin deformation feedback". In: *IEEE transactions on haptics* 8.2 (2015), pp. 209–221.

[96] Alec Rivers, Andrew Adams, and Frédo Durand. "Sculpting by numbers". In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), pp. 1–7.

[97] Alec Rivers, Ilan E Moyer, and Frédo Durand. "Position-correcting tools for 2D digital fabrication". In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pp. 1–7.

[98] Gerald V. Roch, James P. Wiles, and Chris L. Hadley. "Interactive machining system". CA1102434A. June 1981.

[99] L. T. C. Rolt. *A short history of machine tools*. Cambridge, Mass: M.I.T. Press, 1965.

[100] Louis B Rosenberg. “Virtual fixtures: Perceptual tools for telerobotic manipulation”. In: *Proceedings of IEEE virtual reality annual international symposium*. IEEE. 1993, pp. 76–82.

[101] Douglas T Ross. *General Description of the APT system*. Tech. rep. Volume 1. Servomechanisms Laboratory, Massachusetts Institute of Technology, 1959.

[102] Vedant Saran and Eric Paulos. “Turn-by-WiAR: A Computationally-Mediated Approach to Rapid Prototyping of Physical Objects”. MA thesis. EECS Department, University of California, Berkeley, May 2019.

[103] Greg Saul et al. “SketchChair: an all-in-one chair design system for end users”. In: *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 2010, pp. 73–80.

[104] Valkyrie Savage et al. “Makers’ Marks: Physical markup for designing and fabricating functional objects”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 103–108.

[105] Eldon Schoop et al. “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*. 2016, pp. 1607–1614.

[106] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. “The fast research interface for the kuka lightweight robot”. In: *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*. Citeseer. 2010, pp. 15–21.

[107] Adriana Schulz et al. “Interactive design space exploration and optimization for cad models”. In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–14.

[108] Martin Self and Emmanuel Vercruysse. “Infinite variations, radical strategies”. In: *Fabricate 2017 Conference Proceedings*. JSTOR. 2017, pp. 30–35.

[109] *Shaper Tools: Workstation*. URL: `https://www.shapertools.com/workstation/` (visited on 06/24/2021).

[110] Roy Shilkrot et al. “Augmented airbrush for computer aided painting (CAP)”. In: *ACM Transactions on Graphics (TOG)* 34.2 (2015), pp. 1–11.

[111] *ShoulderX: SuitX*. URL: `https://www.suitx.com/shoulderx` (visited on 06/24/2021).

[112] Robert Simunic. “PowerMILL Robot: Simplifying the Complex Issue of Robotic Programming”. In: *Robotic Fabrication in Architecture, Art and Design 2016*. Springer, 2016, pp. 468–472.

[113] Alexander Slocum. “Kinematic couplings: A review of design principles and applications”. In: *International journal of machine tools and manufacture* 50.4 (2010), pp. 310–327.

[114] S. T. Smith. *Flexures: elements of elastic mechanisms*. Amsterdam: Gordon & Breach, 2000.

[115] Hyunyoung Song et al. "ModelCraft: capturing freehand annotations and edits on physical 3D models". In: *Proceedings of the 19th annual ACM symposium on User interface software and technology.* 2006, pp. 13–22.

[116] Ivan E. Sutherland. "Sketchpad: A Man-machine Graphical Communication System". In: *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference.* ACM, 1963, pp. 329–346.

[117] Russell Taylor et al. "A steady-hand robotic system for microsurgical augmentation". In: *The International Journal of Robotics Research* 18.12 (1999), pp. 1201–1210.

[118] Alexander Teibrich et al. "Patching physical objects". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology.* 2015, pp. 83–91.

[119] Andreas Thoma et al. "Robotic fabrication of bespoke timber frame modules". In: *Robotic Fabrication in Architecture, Art and Design.* Springer. 2018, pp. 447–458.

[120] Rundong Tian et al. "Matchsticks: Woodworking through improvisational digital fabrication". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 2018, pp. 1–12.

[121] Rundong Tian et al. "Turn-by-wire: Computationally mediated physical fabrication". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology.* 2019, pp. 713–725.

[122] Maryam Tohidi et al. "Getting the right design and the design right". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems.* 2006, pp. 1243–1252.

[123] Cesar Torres, Wilmot Li, and Eric Paulos. "ProxyPrint: Supporting Crafting Practice through Physical Computational Proxies". In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems.* 2016, pp. 158–169.

[124] Jocelyne Troccaz and Yves Delnondedieu. "Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC)". In: *Mechatronics* 6.4 (1996), pp. 399–421.

[125] Sherry Turkle. *Simulation and its discontents.* MIT press, 2009.

[126] Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. "Tools, Tricks, and Hacks: Exploring Novel Digital Fabrication Workflows on #PlotterTwitter". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 2021, pp. 1–15.

[127] Ryo Umemoto et al. "Work assisting tool". JP2019177431A. Oct. 2019.

[128] Nobuyuki Umetani et al. "Designing Custom-made Metallophone with Concurrent Eigenanalysis." In: *NIME.* Vol. 10. Citeseer. 2010, pp. 26–30.

[129] Lauren Vasey et al. “Behavioral design and adaptive robotic fabrication of a fiber composite compression shell with pneumatic formwork”. In: *Proceedings of the 35th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*. CUMINCAD, 2015.

[130] Matthias Wandel. *Pantorouter*. URL: `https://woodgears.ca/pantorouter/` (visited on 06/24/2021).

[131] Lingfeng Wang and Emily Whiting. “Buoyancy optimization for computational fabrication”. In: *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, pp. 49–58.

[132] Christian Weichel et al. “MixFab: a mixed-reality environment for personal fabrication”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 3855–3864.

[133] Christian Weichel et al. “ReForm: integrating physical and digital design through bidirectional fabrication”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 93–102.

[134] Christian Weichel et al. “SPATA: Spatio-tangible tools for fabrication-aware design”. In: *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. 2015, pp. 189–196.

[135] *What’s the Difference Between Vertical and Horizontal Milling Machines?* URL: `https://www.autodesk.com/products/fusion-360/blog/difference-between-vertical-and-horizontal-milling-machines/` (visited on 06/24/2021).

[136] Karl D. D. Willis et al. “Interactive fabrication: new interfaces for digital fabrication”. In: *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 2010, pp. 69–72.

[137] Karl D. D. Willis et al. “Spatial sketch: bridging between movement & fabrication”. In: *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*. 2010, pp. 5–12.

[138] Xun William Xu and Stephen T Newman. “Making CNC machine tools more open, interoperable and intelligent — a review of the technologies”. In: *Computers in Industry* 57.2 (2006), pp. 141–152.

[139] Junichi Yamaoka and Yasuaki Kakehi. “dePENd: augmented handwriting system using ferromagnetism of a ballpoint pen”. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 2013, pp. 203–210.

[140] Mayu M Yamashita, Junichi Yamaoka, and Yasuaki Kakehi. “Enchanted scissors: A scissor interface for support in cutting and interactive fabrication”. In: *ACM SIGGRAPH 2013 Posters*. 2013, pp. 1–1.

[141] Jiaxian Yao et al. “Interactive design and stability analysis of decorative joinery for furniture”. In: *ACM Transactions on Graphics (TOG)* 36.2 (2017), pp. 1–16.

[142] Hironori Yoshida, Maria Larsson, and Takeo Igarashi. "Upcycling tree branches as architectural elements through collaborative design and fabrication". In: *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 2019, pp. 589–593.

[143] Amit Zoran. "Hybrid basketry: interweaving digital practice within contemporary craft". In: *ACM SIGGRAPH 2013 Art Gallery*. 2013, pp. 324–331.

[144] Amit Zoran and Joseph A Paradiso. "FreeD: a freehand digital sculpting tool". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 2613–2616.

[145] Amit Zoran et al. "The hybrid artisans: A case study in smart tools". In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 21.3 (2014), pp. 1–29.

[146] Amit Zoran et al. "The wise chisel: The rise of the smart handheld tool". In: *IEEE Pervasive Computing* 13.3 (2014), pp. 48–57.