Adroid: Augmenting Hands-on Making with a Collaborative Robot

Rundong Tian
University of California, Berkeley
Berkeley, United States of America
rutian@berkeley.edu

Eric Paulos
University of California, Berkeley
Berkeley, United States of America
rutian@berkeley.edu

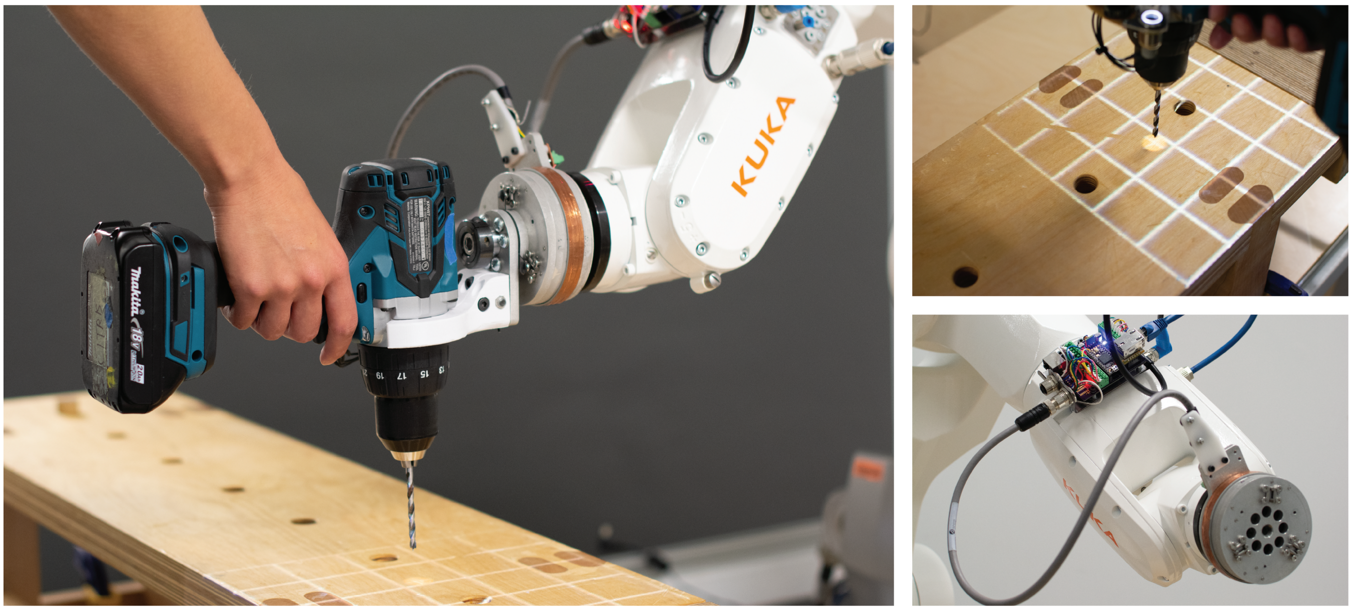

Figure 1: *Adroid* allows users and robots to work collaboratively with hand-held tools. Left: The user has attached a drill to the robot. The user moves the drill by hand, and *Adroid* computationally constrains what motions are possible. Top right: A projection augmented reality interface overlays information about the current tool onto the workpiece. Bottom right: This robot is not backdrivable. A custom force-torque sensor measures the user's forces and is used by *Adroid* to determine how to move the robot.

ABSTRACT

Adroid[1] enables users to borrow precision and accuracy from a robotic arm when using hand-held tools. When a tool is mounted to the robot, the user can hold and move the tool directly—*Adroid* measures the user's applied forces and commands the robot to move in response. Depending on the tool and scenario, *Adroid* can selectively restrict certain motions. In the resulting interaction, the robot acts like a virtual "jig" which constrains the tool's motion, augmenting the user's accuracy, technique, and strength, while not diminishing their agency during open-ended fabrication tasks. We complement these hands-on interactions with projected augmented reality for visual feedback about the state of the system. We show how tools augmented by *Adroid* can support hands-on making and discuss how it can be configured to support other tasks within and beyond fabrication.

[1] *Adroid* is a play on "adroit" and "a droid"; it is a single robot which augments hands-on fabrication.

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10.
https://doi.org/10.1145/3472749.3474749

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Haptic devices.*

KEYWORDS

Digital Fabrication, Interactive Fabrication, Haptics, Robots, Projection AR, Lucid Fabrication

ACM Reference Format:
Rundong Tian and Eric Paulos. 2021. *Adroid*: Augmenting Hands-on Making with a Collaborative Robot. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3472749.3474749

1 INTRODUCTION

Industrial robot arms are highly versatile tools. From free-form 3D printing [6, 25] to timber frame manufacturing [33], researchers have explored many creative applications that leverage the capabilities of these machines. Current digital fabrication workflows for industrial robots share many similarities with the workflows of more commonplace digital fabrication tools, such as 3D printers or CNC routers: a user designs the desired geometry in a computer aided design (CAD) software, develops the corresponding toolpath in a computer aided manufacturing (CAM) software, and sets up the fabrication tool to execute the toolpath [34]. Similar workflows yield similar limitations—there is enormous overhead for articulating a desired geometry to a digital fabrication tool, and that fabrication tool is unaware of discrepancies between the real world where the toolpath is executed and the simulated world where the toolpath was created.

The challenges of navigating these workflows have inspired many projects in *Interactive Fabrication* [43]. Proposed by Willis et al. in 2011, *Interactive Fabrication* tools enable users to bypass conventional workflows and control digital fabrication tools directly through embodied, hands-on interactions. Many related projects developed systems around a single tool, such as a router [29, 47], pen [16, 44], 3D printer [25], laser cutter [20], or lathe [38]. Notably, similar concepts first emerged in the early 1950s with a system called *record-playback*, in which a user could work directly with a manual machine tool (such as a lathe) to fabricate a component, and the system could automatically fabricate replicas of that component by recording and playing back how the user interacted with the tool [23, 24]. *Adroid* shares similar motivations as these other works—how might users gain the benefits associated with digital fabrication tools while avoiding the overhead of conventional workflows? Instead of exploring how individual tools can be augmented, we examine how the user can be augmented directly through a robotic arm (Figure 1).

Our 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 preprogrammed, only constraints—as the user tries to move a tool mounted to the robot, *Adroid* generates motions on the fly in response to the user's 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 feedback about the current tool, such as its position with respect to a workpiece.

As a starting point, we are interested in exploring how virtual jigs can support intermediate and expert woodworkers in shared workshops. Using a tool mounted to *Adroid* requires expertise with the manual version of that tool. Shared workshops—in both professional settings and makerspaces—can democratize expensive machinery like robotic arms.

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.

2 RELATED WORK

2.1 Virtual and physical fixtures

Themes around virtual fixturing were explored by early telerobotics researchers. In 1993, Rosenberg demonstrated a system in which a user controlled a robot using a haptic controller, and various virtual constraints—communicated through haptic feedback—assisted the user with spatial orientation [30]. In a similar vein, the original instantiation of "cobots" was unable to move on its own [26]. Instead, it simply guided an operator's motions by adjusting a steering mechanism that could define a path of least resistance. Another early system adapted this concept for revolute joints; motors were replaced with directional clutches [40]. While these techniques have been investigated within the human robot interaction communities, we apply the concept of virtual fixtures to mediate the use of hands-on fabrication tools.

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 similar jigs can be computationally designed and fabricated on-demand [19]. 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.

2.2 Controlling industrial robots today

At the lowest level, users can program robot motions through 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 input/output 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 [34]. Spurred by developments in filmmaking [1], animation tools like Autodesk Maya have also been co-opted as a way to simulate and generate robotic motions [2]. However, toolpaths generated through these external tools are often static—the robot is oblivious to 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 programming language of the robot by exposing an interface for controlling robot motions [10, 11, 32]. 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 [14] and Quipt [9] 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.

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 [21, 25] as well as multi-axis toolpaths for 3D printing [6, 13]. Segments of a large scale fabrication task were delegated to human and robot workers in [15]. Within the architecture research community, industrial robotic arms are widely used as a platform for exploring new manufacturing materials and workflows [37, 41, 42]. In industry, Hilti recently released a robotic platform that semi-autonomously drills overhead holes for commercial building construction [3].

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 sensing, which allow users to "program" robot positions by directly moving the robot [4, 17]. During these types of interactions, the robot can also communicate additional information through haptics, such as repelling forces near singularities [8]. In the context of virtual reality, researchers have explored how a robot can be used as a general purpose haptic controller for rendering force feedback [7]. Rather than use physical contact only to demonstrate a motion that is ultimately meant to be performed autonomously, our system utilizes continuous physical contact to mediate the direct and collaborative use of hand-held tools.

2.4 Mediating hand tool usage

Researchers frequently seek to augment human physical abilities by computationally mediating the control of hand tools. These aspirations are shared across the domains of digital fabrication, robotic surgery, vehicle control, and more. In digital fabrication research, many variations of physical or visual feedback have been explored. Orthogonal to the type of assistance is the intent of the assistance—does the system guide the user to a specific end result, or facilitate the process of using the tool? For example, visual feedback has been used to communicate a desired outcome with respect to a digital model [28], as well as to show the correct technique for using workshop tools [31]. Direct physical feedback has also been explored. In some work, sensors and actuators actively correct the input of the user as they are using an augmented tool [18, 29, 47], where the target of the correction is a predefined digital model. Alternatively, computationally designed physical jigs offer users physical feedback in the context of woodworking power tools [19], as well as wire [39] and fiber [46] wrapping. In these projects, users are not as strictly bound to a predefined result, and work by Torres et al. directly examines the diverging intent of similar scaffolding [39]. Haptics has also been utilized for physical feedback—projects have used force feedback to guide the use of pens [44], scissors [45], hot-glue guns [18], as well as lathes [38]. Again, the same modality (haptics) was used for diverging goals—for example, in [38] to assist with technique and accuracy, and in [18] to guide users to a predefined model. In this work, we provide direct physical feedback in a way similar to physical jigs, which support users in the process of working with physical tools. Rather than physically fabricate jigs like [19, 39, 46], our robot is programmed to become various jigs.

Outside of digital fabrication, most systems facilitate the *process* of using tools; the idea of a "correct" result in these contexts is more ambiguous. As noted by [48], one especially relevant related domain is medicine, where numerous techniques for augmenting surgical tools have been proposed. For example, researchers have explored how teleoperated surgical systems can facilitate delicate motions by smoothing and magnifying a surgeon's hand gestures [12], "steady" the patients movements by synchronizing the robotic arms to the patient's heartbeat [22], or augment a surgeon's existing sense of touch through haptic feedback via sensory substitution [5]. In contrast, other systems allow a surgeon and a robot to simultaneously hold a tool when performing extremely delicate surgical tasks [36]. 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.

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 that a user exerts on the tool and commands the robot to move accordingly in response. In addition, we provide standard user interface elements for visual feedback and user input. We display spatial information about the system's 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; in our current implementation, we use a tablet to ease how user inputs are captured.

3.1 Four prototype tools

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

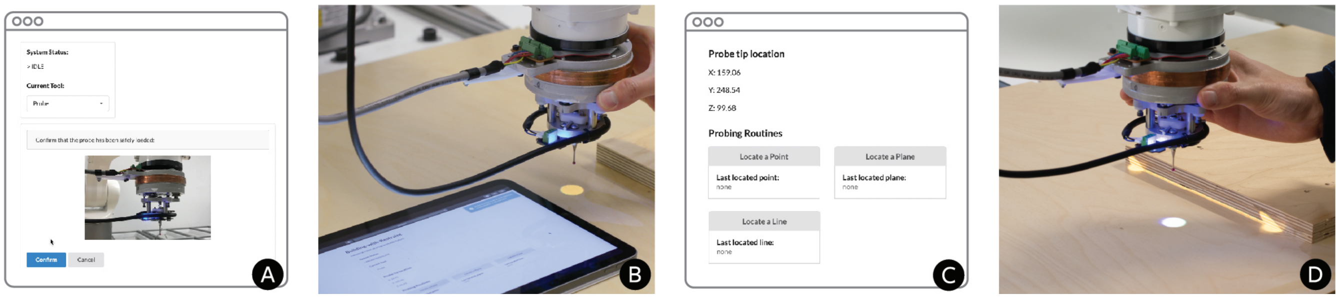

Figure 2: A: The top segment of the tablet GUI contains a dropdown for tool selection, and the bottom segment displays information specific to the current tool. The bottom segment is currently asking the user for confirmation that the probe has been successfully mounted. All following GUI screenshots zoom in on the bottom segment. B: When the tip of the probe contacts an object, the probed location is highlighted by the AR display in orange. C: The GUI for the probe contains buttons that allow the user to locate specific types of geometries, such as a plane or an edge, which require multiple points to be probed. When one of these buttons is pressed, the system prompts the user to probe the necessary series of points to locate that feature. D: After the user has probed an edge, the AR system highlights the probed locations and the found edge in orange. The probe's current location is highlighted in white.

the initial correspondence between the physical and digital models can be established. With the *track saw*, we show how *Adroid* 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.

To use any one of these tools with *Adroid*, the user starts by selecting that tool from the dropdown menu in the tablet GUI (Fig. 2A). Next, the GUI displays instructions to guide the user through connecting the tool. Most tools can be mechanically attached to the robot via a single large thumbscrew. Some tools, such as the probe, also need to be plugged in electronically. After the user connects the tool and confirms the status with the tablet GUI, *Adroid* activates the corresponding constraints and user interfaces for that tool. For more fluid tool changes, we envision that future systems can automatically detect which tool is currently loaded (e.g. via its weight or electrical interface).

Measurement probe — The measurement probe is used to quickly and accurately find the stock material. After the probe has been mounted (Fig. 2A), the user can move it about the workspace by guiding it directly through touch. The probe tool is constrained by the robot to always point downwards; otherwise, the user can move it freely. When the tip of the probe touches an object, the system can measure the probed position (Fig. 2B). To find a geometry which requires multiple points to be probed, such as the edge of a workpiece, the user can press one of the "probing routine" buttons on the GUI (Fig. 2C). For example, when a user clicks on "locate an edge", the GUI will prompt the user to probe two points, the locations of which will be used to calculate the position of the corresponding edge (Fig. 2D). 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, and are commonly used to make straight cuts on planar materials. With this tool, the robot holds the intermediary—the track—rather than the saw (Fig. 3A). When the track is first mounted, the robot constrains the track to maintain its orientation. This way, all cuts that are intended to be parallel will remain parallel as the user moves the track. We incorporated on-tool buttons to minimize the need for a user to switch between the GUI and the tool for frequently used inputs (Fig. 3B). The first button allows the user to lock the track entirely when she is ready to make a cut (Fig. 3C). After making a cut, the user can press this button again to unlock the track. If angled cuts are desired, the second button allows the user to toggle between controlling translation and rotation. In the rotate mode, the system uses the user's applied forces to generate rotational motions. Because all interactions are mediated through the system, a record of how the workpiece was modified (the sequence of locked positions) is generated (Fig. 3F). In other words, when our user interacts with *Adroid* to fabricate an artifact in the real world, she simultaneously creates a lightweight digital model of that object.

Drill — When the drill tool is first mounted, the user can translate the drill, but its orientation is constrained by the robot to point downward (Fig. 4A). The AR system projects a grid onto the workpiece, along with cross-hairs to indicate where the drill is pointing. To assist with precisely positioning holes, snap-to-grid is implemented physically (Fig. 4B). When the user moves the drill near the grid lines, the drill will gently snap into the position of the grid. The system supports two other types of motion for the drill—orientation (Fig. 4D) and drilling (Fig. 4E). Physical buttons next to the drill allow the user to fluidly toggle between these modes, as well as to configure the projected grid (Fig. 4C). The first button lets the user set the current position of the drill's tip as the new origin for the projected grid. The second button allows the user to toggle between controlling translation and orientation. The last latching button lets the user enter and exit the drill mode. When the drill mode is activated, *Adroid* locks the current alignment of the drill and will only allow users to move the tool along the axis of the drill bit. The current system does not allow for the user to configure

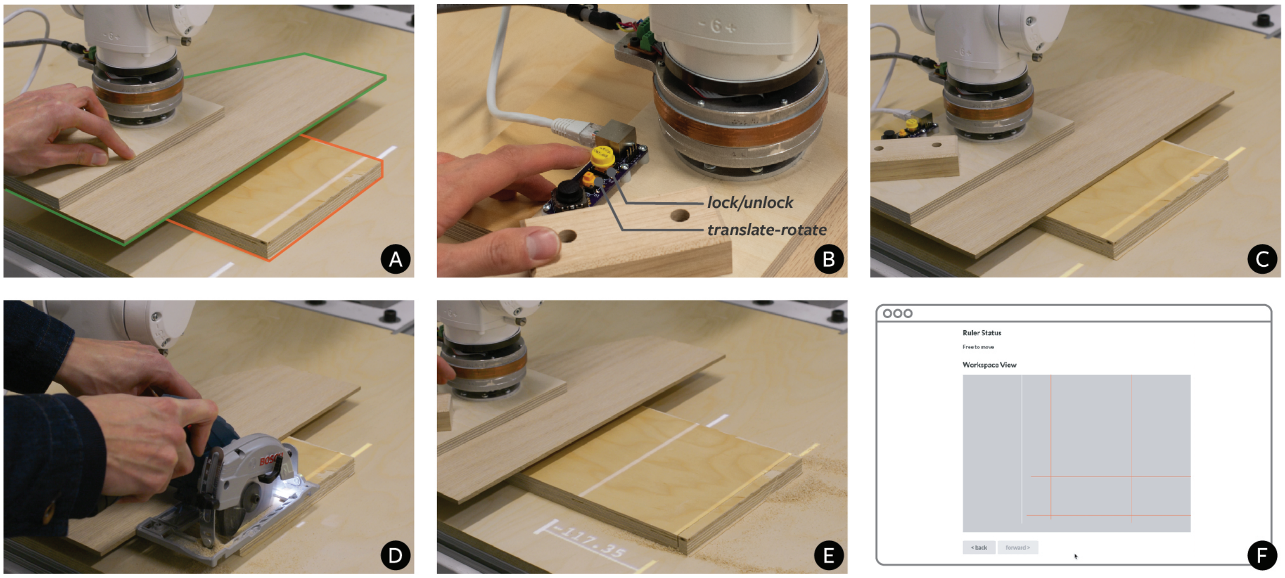

Figure 3: A: For clarity, the track has been outlined in green, and the workpiece in orange. The AR system projects a white line to indicate where the saw will cut relative to the track. B: Two buttons on the track allows the user to lock and unlock the track, and toggle between controlling translation and rotation. C: When the "lock/unlock" button is pressed, the robot lowers the track to the height of the stock material, and the AR interface toggles the saw line color from white to orange. D: With the track locked, the user can use the saw. E: After a cut has been made, the AR system projects the kerf compensated distance between track's current position and the last cut. F: In the GUI, the user can view and navigate the history of previous cuts.

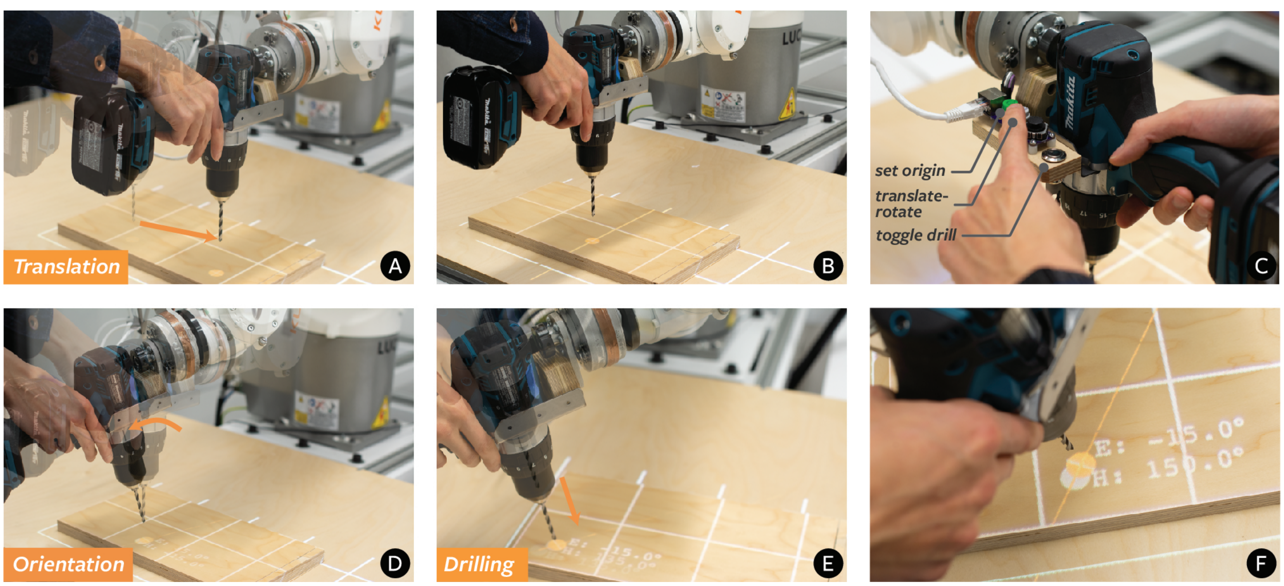

Figure 4: A: For the drill, the AR system projects a grid and crosshairs onto the surface of the table. B: When the user moves the drill near a grid line, the drill will gently snap into the position of the grid. The current grid line is illuminated when the drill is "snapped to" that line. C: Physical buttons on the drill allow users to set the origin of the projected grid, toggle between translation (A) and orientation (D), and 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 of the drill.

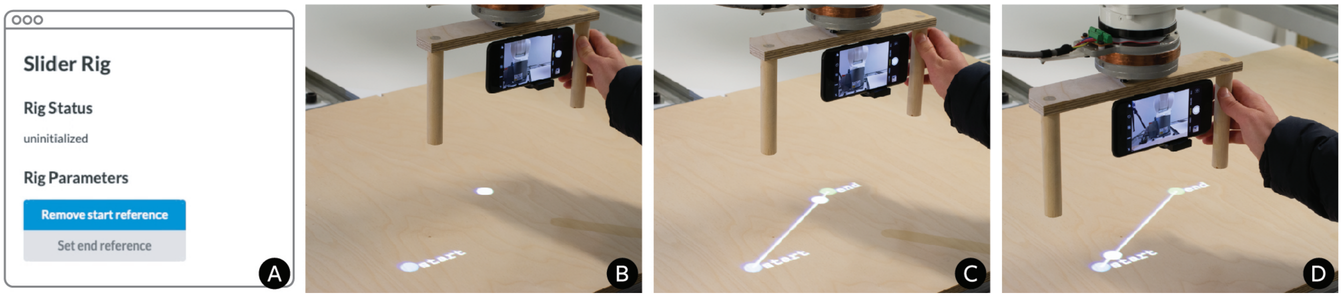

Figure 5: Camera rigs can also be thought of as mechanisms which selectively constrain the motion of a tool. Here, the user defines and uses a virtual slider. A: After moving the camera to a desired start position, the user sets that position using the GUI. B: The projected augmented reality view updates 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 user can move the camera along the virtual slider.

settings such as the drill bit diameter or the spacing of the grid. We envision that the tablet GUI can be used to input these types of settings. Though these details are important, we have currently focused our engineering efforts to demonstrate the core research contribution—programmable constraints for physical fabrication.

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 CAM software is not required.

Camera — 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* is capable of rendering many of these physical jigs through software. As one example, we demonstrate how a user sets up a virtual slider for the camera tool. After the user mounts the camera tool to *Adroid*, the tool can be moved freely. However, the user can further adapt the system such that the camera moves as if were mounted to a slider (Fig. 5). Using the GUI, she can indicate if the camera's 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. This example highlights how *Adroid* allows the user to customize how motions are constrained.

3.2 Proposed scenario: building a step stool

To contextualize how these tools can be leveraged during a fabrication workflow, we walk through a scenario in which our user—an experienced woodworker—is building a wooden step stool: When making a single stool which contains only a few components, she works directly with the tools and materials available at hand rather than draft the entire design on a computer. First, she sketches out a

Figure 6: In a proposed fabrication scenario, we walk through how a user leverages *Adroid* to create this step stool.

design in which the legs are angled outwards (Fig. 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. She fixtures the wood for the seat onto her workbench. She mounts the probe tool, and measures the size and position of the material—the initial geometry is measured, and subsequent operations are contextualized. Next, she removes the probe tool and readies the the track saw. Using the saw, she makes a series of cuts to establish the outer perimeter of the seat. While cutting, she can verify her dimensions with the projection AR interface, which shows the saw's current distance relative to the last cut. No CAD model is required to work directly with this computationally augmented tool; rather, the process of working physically with the tool generates the CAD model. Next, she sets up the drill. After quickly consulting her sketches, she orients the drill at the desired angles, and begins drilling the holes for the legs. Unlike a manual drill press, software can facilitate the setup for this operation. 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 videos of her knolled[2] components to document her process.

4 SYSTEM ARCHITECTURE

Next, we discuss *Adroid's* architecture and highlight elements of the design as they relate to goals such as safety, responsiveness, and modularity.

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

4.1 Industrial robot

In the context of accurately and repeatably fabricating physical objects, the stiffness and accuracy of industrial robots are major assets. We use a Kuka Agilus KR6-R900, an industrial robot with a maximum reach of 900 millimeters and a maximum payload of six kilograms. This robot can be used in CNC applications and has a pose repeatability of 0.02mm. One disadvantage of using robots like the Kuka KR6 is that they are not inherently safe: they are heavy, can move quickly, and are not backdrivable. To safeguard the operator, we limit the maximum speed of the robot to 250 mm/s. In addition, a safety button on the teach-pendant of the robot must be continuously depressed for the robot to move. If the button is released, or pressed too hard, the robot will immediately come to a controlled stop. If the user is operating the robot alone, 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.

4.2 End effector tooling and electronics

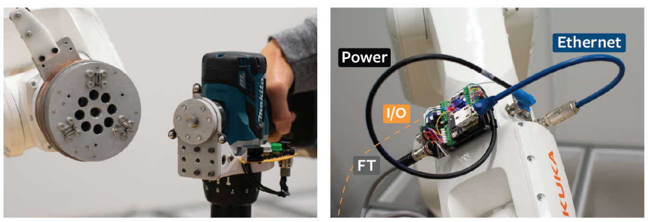

Figure 7: Left: A kinematic coupling allows for tools to be quickly and repeatably mounted. Right: A microcontroller attached to the robot communicates to the force-torque sensor and the end effector tools. The cable to the end effector tools ("I/O") is not shown.

The mechanical interface at the end of the Kuka robot is a mounting flange, which we use to attach the force-torque sensor (Figure 7 left). In order to allow users to quickly and repeatably switch between different tools, we utilize a compliant kinematic coupling as the mechanical interface between the force-torque sensor and each of our tools. Kinematic couplings are known for their high positioning repeatability [35], but are typically ill-suited for high loads. With compliant kinematic couplings, the ball bearings of the kinematic coupling are mounted to flexures, which bend when the coupling is tightened. As a result, the contact points of the kinematic coupling allow for alignment, but are not used to transmit loads.

Electronics on the end effector—e.g. the buttons next to the drill and the force-torque sensor—are connected to an external microcontroller that we mount on the robot (Figure 7 right). The robot has two electrical connectors above its wrist: one for power/inputs/outputs and the other for an internally routed Ethernet cable that terminates at robot's base. The microcontroller uses the former connector for power, and the latter for communicating with the rest of the system.

Next, we highlight implementation challenges and details specific to each tool (Figure 8).

The touch probe's tip is mounted to a spring-loaded kinematic coupling. In this design, the contact points of the kinematic coupling are utilized as rudimentary electronic switches. When the probe tip is displaced, the switches disconnect, allowing the mechanical displacement to be electrically sensed.

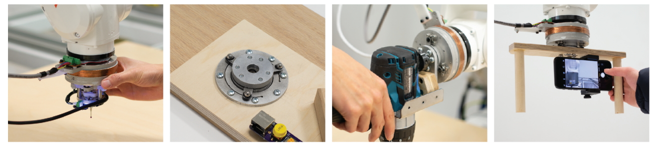

Figure 8: Close up views of each of the tools: probe, ruler, drill, and camera.

The ruler is connected to the robot through a flexural mechanism, which is designed to be compliant out of the plane (to adapt to materials that are not perfectly co-planar with the ruler), but rigid within the plane to ensure accurate cuts.

The drill consists of a battery powered drill attached to our mounting bracket. We selected a drill that can be equipped with 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 drill could rock slightly. We revised the design and fixed the drill to the mounting bracket using the existing fasteners that attach the drill's gearbox to its body.

The camera tool 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.

4.3 Designing and manufacturing a low cost, high resolution force torque sensor

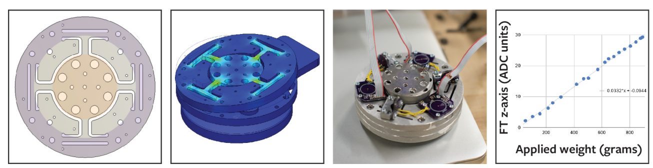

Figure 9: (A) View of the flexure mechanism. The center disk (highlighted orange) is connected to the wrist of the robot, and the outer ring (highlighted purple) is connected to various end effectors. The disk and the ring are joined by thin T-shaped flexures which are designed to elastically deform. (B) FEA studies of the force-torque sensor helps ensure that applied forces and resulting deflections are measurable by our sensors and tolerable by the material. Displacements are exaggerated. (C) Inside view of the force torque during assembly. Each of the printed circuit boards contain a spiral inductor for distance sensing. (D) Plot of Z-axis signal response with respect to applied weight.

The force-torque sensor is what enables users to control tools on the robot through touch. However, commercial force-torque sensors can cost upwards of $5000 USD, which is nearly 20% the cost of the entire robot. In response, we endeavored to create a sensor that was high resolution, easy to manufacture, and low cost. Our 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 (Figure 9 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. We developed this sensor only as far as to support the interactions described in this paper, and additional refinement and characterizations are possible.

Like many existing force-torque sensors, the core of our sensor is a compliant mechanism—a geometry designed to elastically deform under load (Figure 9 left). By measuring how the mechanism deforms, the applied forces can be calculated given the mechanism's stiffness (i.e. $F = kx$). In order to sense displacements, we use three printed circuit board (PCB) spiral inductors, whose effective inductance shifts in response to their proximity to nearby metals (Figure 9 center right). Each PCB inductor is mounted facing a conductive target and is monitored by a 28-bit inductance-to-digital converter (Texas Instruments LDC1612, used in metal proximity sensing). These sensors communicate directly to the microcontroller on the wrist of the robot.

One limitation of inductance sensors is that they are sensitive to strong magnetic fields. While a magnetized screwdriver held directly next to the sensor will noticeably affect the sensor's output, it is not strongly affected by the robot's or drill's motors. We hypothesize that the magnetic flux within those motors are relatively well contained compared to the open field magnet of the screwdriver. Magnetic shielding with high permeability materials such as mu-metal can also be explored.

4.4 Choreographing interactions

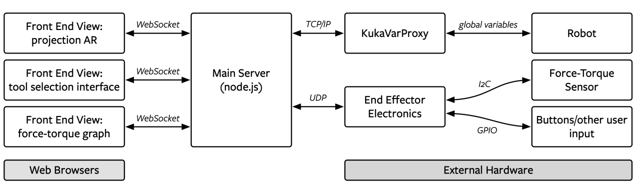

Figure 10: Software architecture. The interactions are controlled by a server communicating to each of the hardware devices (right), as well as the user interface elements (left).

One primary server manages communications between the robot, external hardware, and user interfaces (Figure 10).

Interactive control of an industrial robot — The main server communicates to the robot using KukaVarProxy[3] (KVP), an open source library which establishes a TCP/IP server on the robot's control computer, and allows clients to read and write global variables. We define a set of variables that allow us to communicate desired motions. To control the robot, we run a Kuka Robot Language (KRL) script that continuously reads the status of these variables and moves the robot when appropriate. In our initial test, the motions generated through this scheme were jerky—the KRL script pauses the robot after completing each successive motion. To improve the robot's responsiveness, we incorporated a queue within the KRL script to communicate a short sequence of desired positions. This yielded significantly smoother motions because the robot's motion planner is able to plan the velocities of multiple moves in advance rather than coming to a stop after each move. The main server communicates motions to KVP directly as positions and orientations. Inverse kinematics—solving for the corresponding joint angles—is handled by the robot's controller.

End effector electronics — The Kuka has its own general-purpose input/output (GPIO) connectors on the arm, which can be controlled through KRL. Though it would be possible to use these GPIOs with KRL to monitor some of the electronics (such as the buttons on the tools), we utilize a separate microcontroller for two reasons. First, the IO pins at the robot's wrist are relatively limited (4 inputs and 2 output, all at 24 Volts). Second, we wanted to offload computation away from a low level language like KRL, and coordinate interactions using the main server. Using a separate microcontroller, the main server can communicate to the electronics on the end effector directly, and additional electronics can be more easily incorporated. The only instance where we use one of the robot's GPIO pins (and KRL) is for the touch probe. We use the native GPIO in order to utilize KRL's input interrupts, which can directly report the position at which the probe was triggered.

Graphical user interfaces — GUIs for the tablet and projector are implemented as web applications and communicate to the main server through WebSocket connections. The tablet 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.

4.5 Enabling an augmented drill, software walkthrough

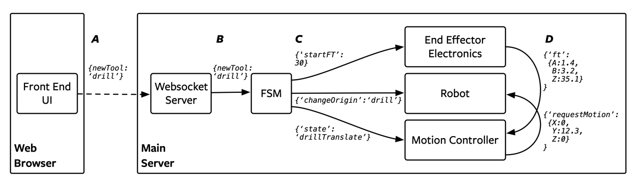

Figure 11: A detailed view into the server which coordinates the interaction: Dotted lines represent communication over WebSocket, solid lines represent communication over the event bus.

To concretize the discussion of the software stack, we step through how the main server handles the drill interactions, from tool selection to generating motions (Figure 11). **(A)** When a user selects the drill tool through the GUI, front-end JavaScript sends a WebSocket message to the main server, which is **(B)** forwarded to finite state machine (FSM) object. **(C)** After the FSM object confirms that this tool change can safely occur, it emits messages to the software modules that interface with the end effector electronics and robot, in order to turn on the force-torque sensor and update the robot's coordinate system. the FSM object then pauses for three seconds to tare the force torque sensor, then **(D)** configures the motion controller object to listen for force-torque messages, calculate the corresponding motion, and relay the appropriate motion command to the robot.

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

5 INTERVIEWS WITH EXPERT FABRICATORS

5.1 Procedure

To probe reactions to our system, we conducted remote semi-structured interviews with four expert fabricators[4]. 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[5]. 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.

5.2 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:

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 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 metalworking.

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.

[4]Due to the COVID-19 pandemic, all user studies were conducted remotely.

[5]The video supplement contains these clips combined into one video.

5.3 Results

Given the small sample size, these interviews were intended to surface insightful perspectives rather than comprehensive findings. Overall, 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 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.

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, we want to create a lightweight way to engage with a set of these capabilities, without necessarily engaging with the entirety of existing workflows.

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, we focused our engineering efforts to demonstrate the core research contribution—using a robot to programmably constrain user motions. In future work, we will refine the interaction details surrounding this core contribution.

6.2 Extending *Adroid* to new tools

In order to integrate a new tool with the current version of *Adroid*, a developer would need to fabricate a mechanical bracket to interface between the new tool and the kinematic coupling, wire any electronic inputs and outputs (such as buttons) to a cable that can plug-in to the microcontroller on the robot, program new virtual constraints and the corresponding logic that will activate these constraints when needed, and implement a GUI to display the relevant parameters and settings for the tool.

However, we envision that *Adroid* can be more easily extended by end-users. For the mechanical design of the bracket, automated techniques leveraging 3D scanning can be explored. To further reduce the amount of custom hardware that users need to fabricated for each new tool, off-the-shelf quick-connect robot tool changers can also be used instead of the kinematic coupling. For the electronics, remappable physical buttons can be incorporated directly on the robot's wrist, such that they can be reused across new tools. For the constraints, users can potentially define and save them through a dedicated GUI. For instance, after the user has mounted the new tool to the robot, a GUI can prompt the user to demonstrate a series of positions of how the tool is allowed to move, and automatically infer the constraint using those positions. A simplified example of this is how the camera tool allows users to customize a linear slider. To allow users to develop the logic of how buttons or a custom GUI would affect the tool's behavior, a variety of techniques from end-user programming, such as visual programming, can potentially be leveraged. Lastly, users could simply download a new virtual jig from an online community and modify it to suit their needs, much like how communities like *Thingiverse* or *GrabCAD* allow members to share, download, and remix mechanical designs.

6.3 Alternative configurations for *Adroid*

Many other robots may be compatible with the interactions enabled by *Adroid*. It is not necessarily the case that the robot must have extremely high strength or stiffness. For example, a smaller robot like the Universal Robotics UR3, which has a 3 kg maximum payload, may not be able to hold a drill, but can hold a drill bushing that offers similar interactions such as snap-to-grid, or rapidly setting angles.

Different force-torque sensors and configurations can also be explored. For example, force-torque sensing can be placed at different locations between the robot and the user. In the current version of the system, the user's applied forces are sensed at the interface between the robot and the tool. One downside of this approach stems from the mechanical compliance of force-torque sensors. Though the force-torque sensor only deflects minimally under load, small angular displacements are amplified by the distance between the sensor and the tool's tip. Alternatively, force sensing can be placed between the tool and the user. In this configuration, the tool will be more rigidly coupled to the robot, and whichever force-torque sensor is chosen no longer needs to structurally support the tool. However, this configuration may require a separate force-torque sensor for each tool.

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.

6.4 Cost

The Kuka robot used in this project costs approximately $30,000 USD. This price places it alongside more expensive fabrication equipment, such as large format CNC routers. In the near future, we envision how shared workshops, such as makerspaces, can make these robots available to many users. Robots designed for domestic tasks offer a hopeful parallel in terms of future cost reductions: the Willow Garage PR2 robot cost around $400,000 USD in 2011, while the recently released Hello Robot is priced at around $20,000 USD.

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, such as the ability to ride a bicycle, is knowledge that is easily known but difficult to communicate [27]. 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, *Adroid* is uniquely positioned to record and playback repeated tasks, such as the sharpening scenario posed by the luthier.

8 CONCLUSION

Our system 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, without traversing the existing workflows required to engage with these tools. In this paper, we 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; and its settings can be instantly saved, restored, and shared. Though we where initially inspired by the unique affordances of physical jigs and futures, this work leverages the concept of a digital jig further as a means to incorporate digital editing affordances directly within physical fabrication.

ACKNOWLEDGMENTS

We thank the Jacobs Institute for Design Innovation for access to the Kuka robots. We also thank Sarah Sterman, Molly Nicholas, Katherine Song, Jeremy Warner, and our interviewees for their insightful feedback. Lastly, we thank the the anonymous reviewers for their constructive comments.

REFERENCES

[1] 2013. *Box - YouTube.* Retrieved 2020-07-01 from https://www.youtube.com/watch?v=lX6JcybgDFo
[2] 2018. Mimic. Retrieved 2020-07-01 from https://www.mimicformaya.com
[3] 2020. Introducing a new construction robot. Hilti Jaibot for construction automation of overhead drilling. Retrieved 2020-03-25 from https://www.youtube.com/watch?v=ytpzbOtNo3w
[4] 2020. KUKA ready2_pilot. Retrieved 2020-03-25 from https://www.youtube.com/watch?v=8yGaMFDuFbE
[5] Brian T Bethea, Allison M Okamura, Masaya Kitagawa, Torin P Fitton, Stephen M Cattaneo, Vincent L Gott, William A Baumgartner, and David D Yuh. 2004. Application of haptic feedback to robotic surgery. *Journal of Laparoendoscopic & Advanced Surgical Techniques* 14, 3 (2004), 191–195.
[6] Chengkai Dai, Charlie CL Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-free volume printing by multi-axis motion. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.
[7] S Devine, K Rafferty, and D Ferguson. 2017. HapticVive—a point contact encounter haptic solution with the HTC VIVE and Baxter robot. *The 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2016 in co-operation with EUROGRAPHICS* (2017), 17–23.
[8] Fotios Dimeas, Vassilis C Moulianitis, Charalampos Papakonstantinou, and Nikos Aspragathos. 2016. Manipulator performance constraints in cartesian admittance control for human-robot cooperation. In *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 3049–3054.
[9] Madeline Gannon. 2015. Quipt — ATONATON. Retrieved 2020-07-01 from https://atonaton.com/quipt
[10] Madeline Gannon. 2015. Robo.Op. Retrieved 2020-07-01 from http://www.madlab.cc/robo-op
[11] Jose Luis García del Castillo y López. 2019. Machina. NET: A Library for Programming and Real-Time Control of Industrial Robots. *Journal of Open Research Software* 7 (2019), 27.
[12] Gary S Guthart and J Kenneth Salisbury. 2000. The Intuitive^{TM} telesurgery system: overview and application. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation.* 618–621.
[13] Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. Framefab: Robotic fabrication of frame shapes. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–11.
[14] Torsten Kröger. 2011. Opening the door to new sensor-based robot applications—The Reflexxes Motion Libraries. In *2011 IEEE International Conference on Robotics and Automation.* IEEE, 1–4.
[15] Benjamin Lafreniere, Tovi Grossman, Fraser Anderson, Justin Matejka, Heather Kerrick, Danil Nagy, Lauren Vasey, Evan Atherton, Nicholas Beirne, Marcelo H Coelho, et al. 2016. Crowdsourced fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology.* 15–28.
[16] Thomas Langerak, Juan José Zárate, Velko Vechev, David Lindlbauer, Daniele Panozzo, and Otmar Hilliges. 2020. Optimal control for electromagnetic haptic guidance systems. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 951–965.
[17] Nicolas Lauzier, Simon Lefrançois, David Castonguay, Louis-Alexis Allen Demers, Jean-françois Duval, Yan Drolet Mihelic, Pierre Olivier Proulx, Rachid Bekhti, Philippe Cardou, Vincent Duchaine, et al. 2017. Force/torque sensor, apparatus and method for robot teaching and operation. US Patent 9,696,221.
[18] Joong Han Lee. 2012. Haptic Intelligentsia. https://vimeo.com/32538081
[19] Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–12.
[20] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the*

25th annual ACM symposium on User interface software and technology. 599–606.
[21] Stefanie Mueller, Anna Seufert, Huaishu Peng, Robert Kovacs, Kevin Reuss, François Guimbretière, and Patrick Baudisch. 2019. FormFab: continuous interactive fabrication. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 315–323.
[22] Yoshihiko Nakamura, Kousuke Kishi, and Hiro Kawakami. 2001. Heartbeat synchronization for robotic cardiac surgery. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Vol. 2. IEEE, 2014–2019.
[23] David F. Noble. 1986. *Forces of production: a social history of industrial automation*. Oxford University Press, New York.
[24] L. R. Peaslee. 1953. Tape-controlled machines. *Electrical Manufacturing* 52 (Nov. 1953), 102–108.
[25] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive fabrication with augmented reality and a robotic 3D printer. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
[26] Michael A Peshkin, J Edward Colgate, Wit Wannasuphoprasit, Carl A Moore, R Brent Gillespie, and Prasad Akella. 2001. Cobot architecture. *IEEE Transactions on Robotics and Automation* 17, 4 (2001), 377–390.
[27] Michael Polanyi. 1967. *The tacit dimension*. Routledge & K. Paul, London.
[28] Alec Rivers, Andrew Adams, and Frédo Durand. 2012. Sculpting by numbers. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–7.
[29] Alec Rivers, Ilan E Moyer, and Frédo Durand. 2012. Position-correcting tools for 2D digital fabrication. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–7.
[30] Louis B Rosenberg. 1993. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE virtual reality annual international symposium*. IEEE, 76–82.
[31] Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. Drill Sergeant: Supporting physical construction projects through an ecosystem of augmented tools. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1607–1614.
[32] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. 2010. 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)*. 15–21.
[33] Martin Self and Emmanuel Vercruysse. 2017. Infinite variations, radical strategies. In *Fabricate 2017 Conference Proceedings*. JSTOR, 30–35.
[34] Robert Simunic. 2016. PowerMILL Robot: Simplifying the Complex Issue of Robotic Programming. In *Robotic Fabrication in Architecture, Art and Design 2016*. Springer, 468–472.
[35] Alexander Slocum. 2010. Kinematic couplings: A review of design principles and applications. *International journal of machine tools and manufacture* 50, 4 (2010), 310–327.
[36] Russell Taylor, Pat Jensen, Louis Whitcomb, Aaron Barnes, Rajesh Kumar, Dan Stoianovici, Puneet Gupta, ZhengXian Wang, Eugene Dejuan, and Louis Kavoussi. 1999. A steady-hand robotic system for microsurgical augmentation. *The International Journal of Robotics Research* 18, 12 (1999), 1201–1210.
[37] Andreas Thoma, Arash Adel, Matthias Helmreich, Thomas Wehrle, Fabio Gramazio, and Matthias Kohler. 2018. Robotic fabrication of bespoke timber frame modules. In *Robotic Fabrication in Architecture, Art and Design*. Springer, 447–458.
[38] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-wire: Computationally mediated physical fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 713–725.
[39] Cesar Torres, Wilmot Li, and Eric Paulos. 2016. ProxyPrint: Supporting Crafting Practice through Physical Computational Proxies. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. 158–169.
[40] Jocelyne Troccaz and Yves Delnondedieu. 1996. Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC). *Mechatronics* 6, 4 (1996), 399–421.
[41] Lauren Vasey, Ehsan Baharlou, Moritz Dörstelmann, Valentin Marshall Prado, Gundula Schieber, Achim Menges, Jan Knippers, et al. 2015. 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)*.
[42] Frédéric Waimer, Riccardo La Magna, Steffen Reichert, Tobias Schwinn, Achim Menges, and Jan Knippers. 2013. Integrated design methods for the simulation of fibre-based structures. (2013).
[43] Karl DD Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D Gross. 2010. Interactive fabrication: new interfaces for digital fabrication. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 69–72.
[44] Junichi Yamaoka and Yasuaki Kakehi. 2013. dePENd: augmented handwriting system using ferromagnetism of a ballpoint pen. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 203–210.
[45] Mayu M Yamashita, Junichi Yamaoka, and Yasuaki Kakehi. 2013. Enchanted scissors: A scissor interface for support in cutting and interactive fabrication. In *ACM SIGGRAPH 2013 Posters*. 1–1.
[46] Amit Zoran. 2013. Hybrid basketry: interweaving digital practice within contemporary craft. In *ACM SIGGRAPH 2013 Art Gallery*. 324–331.
[47] Amit Zoran and Joseph A Paradiso. 2013. FreeD: a freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2613–2616.
[48] Amit Zoran, Roy Shilkrot, Pragun Goyal, Pattie Maes, and Joseph A Paradiso. 2014. The wise chisel: The rise of the smart handheld tool. *IEEE Pervasive Computing* 13, 3 (2014), 48–57.