Open Source, Modular, Customizable, 3-D Printed Kinesthetic Haptic Devices*

Melisa Orta Martinez[1], Joseph Campion[1], Tara Gholami[1], Michal K. Rittikaidachar[1], Aaron C. Barron[1], and Allison M. Okamura[1]

***Abstract*— Open Source Hardware allows users to share, customize, and improve designs, thus enabling technological advancement through communities of practice. We propose open source hardware for educational haptics that permits researchers, educators, and students to share designs arising from their different perspectives, with the potential to expand educational applications. In this paper we present a family of open source kinesthetic haptic devices that build upon the design of a previous educational haptic device, Hapkit 3.0. First, we discuss methods for Hapkit personalization and customization that can be achieved by K-12 students and educators. Next, we describe two kinesthetic haptic device designs that evolved from the original Hapkit 3.0. One uses two standard Hapkits with additional components to form a Pantograph mechanism, and the other uses customized Hapkit elements along with a novel kinematic design to form a serial mechanism. These designs are modular; after building two Hapkits, a user acquires a small number of additional parts to transform them into a two-degree-of-freedom device. The Pantograph mechanism was used in an undergraduate class to teach robotics and haptics to both engineering and non-engineering students. Open source designs for all devices as well as tutorials for customization are available at `http://hapkit.stanford.edu`.**

I. Introduction

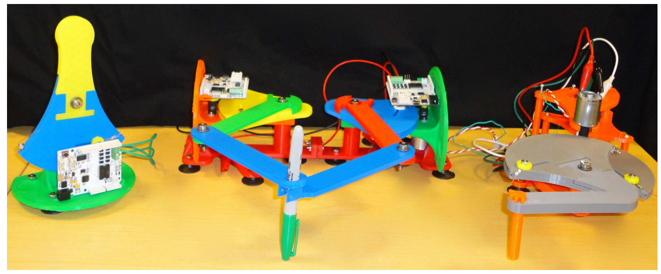
"Open Hardware is a thing – a physical artifact, either electrical or mechanical – whose design information is available to, and usable by, the public in a way that allows anyone to make, modify, distribute and use that thing." [1]

A. Motivation

The Open Source and Free (as in "Libre") software movements have aided in academic research, changed the way technology companies do business, and resulted in products like GNU/Linux, Apache and R [2]–[4]. Open source and Free hardware designs have also had significant impact on technology development. Recent examples include open electronic platforms such as Arduino [5], open 3-D printing solutions such as RepRap [6], and robotics projects [7]. However, with the increased development of proprietary design tools and complex manufacturing processes, open source hardware has not seen the same growth as open source software [2], [4], [8]–[10]. Similarly, most haptic hardware available outside the research community is proprietary. Thus, users have limited freedom to obtain, modify, and learn form these devices. Hardware is more difficult to replicate and modify than software, but increasing availability of 3-D printers, free online Computer Aided Design (CAD) tools, and open electronics platforms enable haptic devices (Figure 1) that are easily manufacturable and modifiable by users inside and outside the research community.

*This work was supported in part by National Science Foundation grant 1441358.

[1]Department of Mechanical Engineering, Stanford University, Stanford CA 94305 USA. `meliso@stanford.edu; aokamura@stanford.edu`



(a) (b) (c)

Fig. 1. Three examples of open source, modular, customizable, 3-D printed kinesthetic haptic devices. (a) Hapkit 3.0: A 1-DOF kinesthetic haptic device [11]. (b) Graphkit: A 2-DOF kinesthetic haptic device comprising two Hapkits connected by a Pantograph mechanism [12]. (c) Haplink: A 2-DOF kinesthetic haptic device that connects two Hapkits in series to form a novel mechanism.

B. Prior Work

One of the first open source kinesthetic haptic devices was the Haptic Paddle [13], a one-degree-of-freedom (1-DOF) haptic device designed at Stanford University for use in an undergraduate course in dynamic systems. Since the release of the Haptic Paddle in 1997, numerous other universities have made their own version of the device [11], [14]–[22], each one with specific improvements. The most recent design from our group, Hapkit 3.0 [11], uses 3-D printed structural components, a low-cost motor and an open source electronic interface based on the Arduino Uno called the Hapkit Board.

Higher-degree-of-freedom kinesthetic devices, as well as tactile devices, have also been made open source. In the early 2000's Campion et al. [23] redesigned the 1994 Pantograph [12] and made it open-source in software and hardware. In 2016 Gallacher et al. [24] made a more accessible Pantograph device by designing an open architecture electronics board and releasing designs using different structural materials. Other examples of open hardware haptic devices include Wooden Haptics [25], the Box and iTouch [26], the Twiddler [27], the Plank [28], and the Tpad Tablet [29]. These open source devices have advanced haptic technology through the implementation of accessible haptic devices, and many of these devices have also been used for education.

Our previous work developing Hapkit [11] and using it in educational environments revealed that a 1-DOF device can facilitate hands-on learning, but many of the two-degree-of-freedom devices cited above have demonstrated a wider range of applications. More complex concepts that naturally follow the topics introduced by Hapkit can only be interactively taught to students if the capabilities of the kit are expanded. This paper describes how Hapkit 3.0 can be evolved/customized into higher-DOF devices.

C. Contributions

In this paper we propose a methodology to customize Hapkit designs, and present two open source designs that expand the capabilities of the original Hapkit to 2-DOF (Figure 1). Both 2-DOF devices aim to conserve the robust design, low cost, and ease of manufacturing with readily accessible tools of the original Hapkit. The aim is to provide resources and a process by which users can build and configure a wide range of haptic devices. The intended audience includes researchers, educators, and students who are interested in building open source inexpensive haptic devices as well as use haptic devices for educational applications.

II. Personalization and Customization

Here we distinguish between *personalization* and *customization* because of their different goals and constraints. In our work, personalization allows users to change the device cosmetically and feel an increased sense of ownership. This is appropriate for users (such as K-12 educators and students) who may lack the technical expertise required for more comprehensive, functional design changes, but would benefit educationally from using CAD tools and 3-D printing, as well as the engagement we observed when students make their own devices [11]. In contrast, customization allows users to create devices with new functionality. This is appropriate for designers who have access to more advanced design and manufacturing tools.

A. Personalization

The ability to manufacture and assemble a functional device such as Hapkit allows the teaching of fundamental concepts, such as kinematics and physical mechanics, through interactions with a physical device. Yet the software used by haptics researchers for device development and customization is typically too costly for K-12 educators and students. The stereolithography (.stl) files created for 3-D printing of Hapkit 3.0 were made with SolidWorks, a licensed CAD design software. For that reason, previous classroom dissemination of Hapkit used completely pre-designed devices.

Preliminary work with undergraduate students indicated that personalization causes users to feel more engaged with their use of Hapkit [11], so we developed a method for users to personalize their device using SketchUp, a 3-D modeling software that is free for K-12 students and educators. To prevent changes to the *functional* components of Hapkit 3.0 and 2-DOF versions of Hapkit (Graphkit and Haplink described below), personalization focused only on the handle of the device, where the user grasps the device. The original circular design of Hapkit's handle is not related to the device's functionality.

A tutorial for personalization was written and disseminated on the Hapkit website. To enable personalization in Sketchup, we took various handle shapes originally created in SolidWorks, converted them to .stl files, and imported those files to SketchUp. In Sketchup, users can select one of these shapes and add their own design on the surface of the handle. For example, using SketchUp's 3-D text tool, a student can add his or her name to the front of the handle and using the various drawing tools, users also have the ability to create their own personal design below their name. More advanced users can also modify the original circular handle shape by sketching a new handle design onto the original circular layout. Then using the push/pull tool, users can extrude their design and export their SketchUp files to .stl files, which can be used for 3-D printing. The user instructions as well as basic starting designs can be found at `http://hapkit.stanford.edu/build.html`.

B. Customization

Our design goals for 2-DOF kinesthetic haptic devices are generated from both manufacturing/assembly requirements and dynamics/rendering requirements. We aimed for "Open Implementation" (the design should be available and free to use and modify [10]) and accessibility (users should be able to obtain or make the parts using commonly available tools and online purchasing). In addition, we desired modular higher-DOF designs that use the original Hapkit 3.0 as a component. In doing so, we maintain one of the core ideas of Hapkit, that it is a "kit" to be assembled. Therefore, we also designed for assembly, such that the device is composed of a minimal number of parts and that the assembly process is robust and easy to explain. The 2-DOF impedance-type haptic devices must be capable of rendering compelling virtual environments. To date we do not know what level of kinesthetic device performance is required for effective application in education, but we use general haptic device guidelines to define goals such as backdrivability, providing users with a consistent feel throughout the workspace, a large Z-Width [30], and a maximum stable stiffness of at least several hundred N/m. We also require our devices to have a maximum force output on the order of 5 N (for safety), high-resolution position sensing (using a magnet and magneto-resistive sensor), accurate torque output, and the ability to close a haptic control loop at a rate of at least several hundred Hz [31]–[34]. These goals are addressed in Sections III-C and IV-C.

III. Graphkit

A. Design

Graphkit was originally designed with the aim of turning Hapkit 3.0 into a 2-DOF haptic device that could also be used as a programmable drawing tool for a robotics class. A Pantograph design, a kinematically well conditioned device [23], was chosen because its planar workspace is practical for drawing (Figure 2).

Graphkit takes advantage of Hapkit 3.0's modular design by reusing the capstan drive mechanisms, and allowing students to expand the kinematics learned for 1-DOF to 2-DOF. Graphkit is made from two original Hapkits 3.0 and a few additional parts (Figure 3), resulting in additional cost of approximately $10 USD per kit. Because Graphkit's design requires mirrored components, the additional parts kits are divided into "left kits" and "right kits". Two students, one with a "left kit" and one with a "right kit" who have each previously made their own 1-DOF device, can work together to make a Graphkit. Students first construct the base by attaching its left and right sides together. Then the two Hapkits are modified by replacing their handles with the Handles for Graphkit, and laying them with the Sector Pulley parallel to the table in order to attach them to the new bases. Finally, the two remaining links are attached to the Hapkits' new handles. In order to electrically connect the two Hapkits, one Hapkit board is chosen to be the "Master" and the other the "Slave". The "Slave" board is powered but runs no code in the microprocessor. (This powers the magnetoresistive sensor whose output is read by the "Master" board in an analog input.) The "Master" runs all the control software and takes the input from both magnetoresistive sensors to control both motors. The "Master" and "Slave" boards are connected using alligator clips as shown in Figure 2(a).

B. Software and Control

In order to render virtual environments using Graphkit, we implement a control loop on the "Master" Hapkit Board using the Arduino programming language. We obtain the position of the end-effector by reading the angles θ_1 and θ_2 using the magnetoresistive sensors on both Hapkit Boards in the main software loop. Using the forward kinematics illustrated in Figure 2(b) and following [23], the position of the end-effector $P_3(x_3, y_3)$ is given by

$$x_3 = x_\mathrm{h} + \frac{\|P_3 - P_\mathrm{h}\|}{\|P_2 - P_4\|}(y_4 - y_2) \tag{1}$$

$$y_3 = y_\mathrm{h} + \frac{\|P_3 - P_\mathrm{h}\|}{\|P_2 - P_4\|}(x_4 - x_2) \tag{2}$$

where

$$P_2(x_2, y_2) = [l_2 cos(\theta_1), l_2 sin(\theta_1)] \tag{3}$$

$$P_4(x_4, y_4) = [l_5 cos(\theta_2), l_5 sin(\theta_2)] \tag{4}$$

$$\|P_2 - P_\mathrm{h}\| = \frac{(l_4^2 - l_3^2 + \|P_4 - P_2\|^2)}{2\|P_4 - P_2\|} \tag{5}$$

$$P_\mathrm{h} = P_2 - \frac{\|P_2 - P_\mathrm{h}\|}{\|P_2 - P_4\|}(P_2 - P_4) \tag{6}$$

$$\|P_3 - P_\mathrm{h}\| = \sqrt{l_3^2 - \|P_2 - P_\mathrm{h}\|^2} \tag{7}$$

We calculate a desired vector force, F, depending on the virtual environment being rendered. The desired output torque

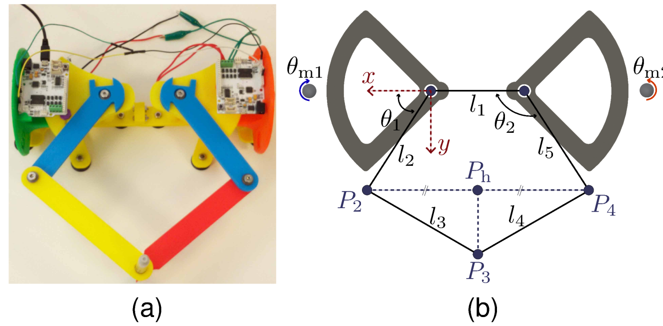


Fig. 2. Graphkit: A 2-DOF kinesthetic haptic device based on the Pantograph mechanism and made of 3-D printed structural components. (a) Built Graphkit. (b) The Kinematic model of Graphkit. θ_1 corresponds to the angle of rotation of the link on the side with the Master Hapkit Board, and θ_2 corresponds to the angle of rotation of the link on the other side. P_i is the position of joint i, where P_3 is the position of the end effector and P_h is the point of intersection between segments P_2P_4 and the height of triangle $P_2P_3P_4$.

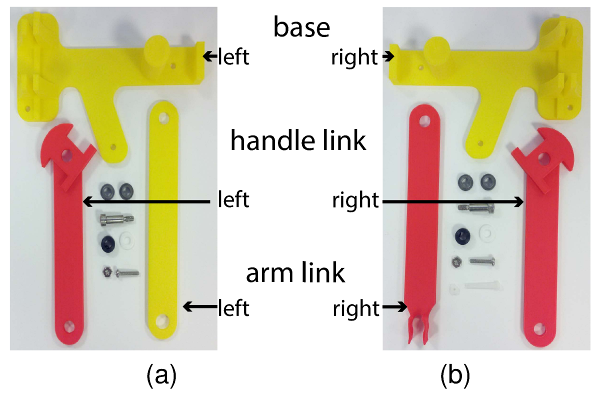


Fig. 3. Additional parts beyond Hapkit 3.0 required to construct Graphkit. (a) The "left kit". (b) The "right kit".

of the motors is:

$$T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = J^\mathsf{T} F \tag{8}$$

where J is the Jacobian from [23].

C. Device Capabilities

Graphkit extends the capabilities of a single 1-DOF Hapkit to a planar 2-DOF device. Detailed performance quantification is left for future work, but we report here that the device is capable of rendering various virtual environments within its workspace illustrated in Figure 4(a) and it can close the haptic control loop on the Hapkit Board at approximately 120 Hz while rendering a 2-DOF virtual environment. With the current link lengths and the use of the same Mabuchi motor as in the original Hapkit 3.0, Graphkit can output a maximum force of 3 to 4 N depending on the position of the end-effector in the workspace.

IV. Haplink

A. Design

Figures 5 and 6 show Haplink, another planar 2-DOF haptic device created as a customization project for Hapkit

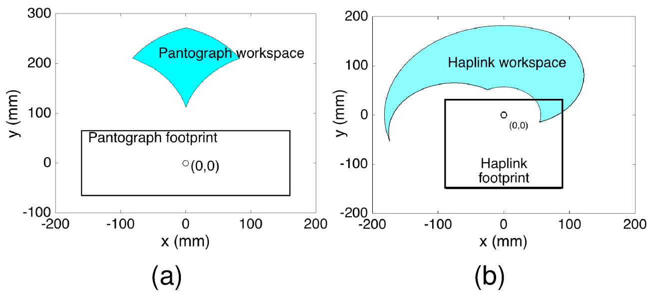


Fig. 4. Workspace achieved by (a) Graphkit with $l_1 = 88.5$ mm, $l_2 = 126$ mm, $l_3 = 152.4$ mm, $l_4 = 152.4$ mm, and $l_5 = 126$ mm and (b) Haplink with $l_a = 83$ mm, $l_b = 100$ mm, $\Delta\theta_a = 50$ deg and $\Delta\theta_b = -80$ deg.

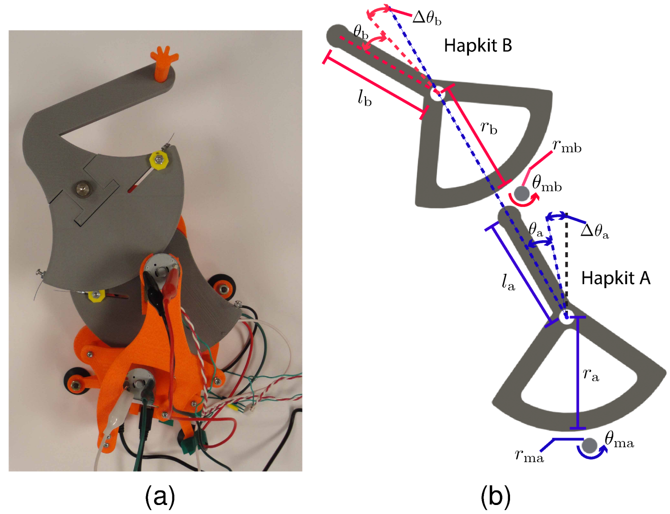


Fig. 5. Haplink: A 2-DOF kinesthetic haptic device that uses a coupled serial drive mechanism and 3-D printed structural components. (a) Built Haplink. (b) Kinematic model of Haplink. l_a is the distance between the center of rotation of Hapkit A and the center of rotation of Hapkit B. l_b is the distance between the center of rotation of Hapkit B and the end-effector. r_a and r_b are the radii of Hapkits A and B respectively. r_{ma} and r_{mb} are the radii of motors A and B respectively. θ_{ma} and θ_{mb} are the angles of rotation of motors A and B. θ_a and θ_b are the angles of rotation of Hapkits A and B. $\Delta\theta_a$ and $\Delta\theta_b$ are the initial offset angles of the handles of Hapkits A and B, respectively.

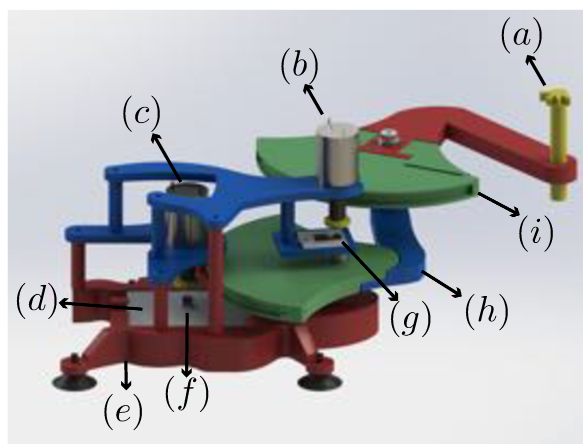


Fig. 6. Diagram showing the components of Haplink: (a) User Handle, (b) Motor B, (c) Motor A, (d) Hapkit Board, (e) Base, (f) Magnetoresistive sensor A, (g) Magnetoresistive sensor B, (h) Hapkit A, (i) Hapkit B.

but not restricted to use unmodified parts. The device is composed of two Hapkit Sector Pulleys with modified handles connected in series using a novel mechanism that allows both motors to be grounded (Figure 6) by a modified base and suction-cup mounts. Unlike other two-degree-of-freedom serial chain mechanisms that require pulleys or cables that apply off-axis forces on the motors, Haplink allows direct transmission from two grounded motors by capstan drives.

Haplink maintains major aspects of the 3-D printed structural components and the ability of users to design their own end-effectors. Starting from a single Hapkit 3.0 device, the additional hardware required includes 3-D printed components, a motor ($3.50), a magnetoresistive sensor ($7.00), screws, and a shaft collar. As with Hapkit, Haplink uses magnetoresisitive sensors (KMA210) positioned under magnets attached to the shaft of motors (Mabuchi) to measure the rotation of the motors which drive the device. The magnetoresistive sensors are connected to a single Hapkit board, which drives both motors. Figure 5(b) shows a model of the device. We refer to the first Hapkit Sector Pulley in the series as Hapkit A and the second one as Hapkit B. Hapkit A rotates about a pivot point fixed in space. The pivot point of Hapkit B is located on the end-effector of Hapkit A, such that the frame of Hapkit B rotates with Hapkit A. The motor driving Hapkit B is grounded and coaxial with the pivot point of Hapkit A, which eliminates any reaction torques from motor B onto Hapkit A. This also ensures that the pivot point of Hapkit B is always the same distance from motor B even though Hapkit B's frame moves with Hapkit A. In order to control Haplink, we use our original Hapkit Board located with Hapkit A and we added one magnetoresistive sensor, grounded and located under the magnet attached to the shaft of motor B.

Similarly to Graphkit, Haplink has low inertia because both motors are grounded, preventing the highest mass components from moving. Haplink also maintains low friction by using the capstan drive transmission from Hapkit 3.0's design [11]. Additionally, we maximized the device's workspace (shown in Figure 4(b)) and minimized the length of the handles (l_a and l_b) by optimizing the handle's angles ($\Delta\theta_a$ and $\Delta\theta_b$) and the rotation angle of the device while constraining ourselves to the following: the workspace should be centered in front of the device's footprint, the radius of the Sector Pulleys should be consistent with Hapkit 3.0, and a square with an area of at least 100 cm^2 should fit in the workspace. Our final design increased the rotation angle of the Sector Pulleys from 90 to 120 degrees compared to previous Hapkit designs. The dimensions of the final configuration are given in Figure 4(b).

B. Software and Control

Haplink is controlled with a single Hapkit Board programmed using the Arduino programming language. In order to render virtual environments, the control loop computes an output force based on the end-effector position, and this is used to compute torques at the two motors. We measure θ_a and θ_b by measuring the output from the magnetoresistive sensors in an interrupt triggered at a rate of 1 kHz and keeping track of the flips of the magnet. We then compute θ_{ma} and θ_{mb} in the main loop using:

$$\theta_{ma} = -\frac{r_a}{r_{ma}}\theta_a \qquad (9) \qquad \theta_{mb} = -\frac{r_b}{r_{mb}}\theta_b \qquad (10)$$

The forward kinematics are:

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} -l_a \sin(\tilde{\theta}_a) - l_b \sin(\tilde{\theta}_a + \tilde{\theta}_b) \\ l_a \cos(\tilde{\theta}_a) + l_b \cos(\tilde{\theta}_a + \tilde{\theta}_b) \end{bmatrix} \tag{11}$$

where:

$$\tilde{\theta}_a = \theta_a + \Delta\theta_a \tag{12}$$

$$\tilde{\theta}_b = \theta_b + \Delta\theta_b \tag{13}$$

The desired vector force F is calculated depending on the virtual environment being rendered. The desired torque at the motors is then:

$$T_m = \begin{bmatrix} T_{ma} \\ T_{mb} \end{bmatrix} = J^T F \tag{14}$$

where:

$$J = \begin{bmatrix} \frac{r_{ma}}{r_a}(l_b \cos(\tilde{\theta}_{ma} + \tilde{\theta}_{mb}) + l_a \cos(\tilde{\theta}_{ma})) & \frac{r_{mb}}{r_b} l_b \cos(\tilde{\theta}_{ma} + \tilde{\theta}_{mb}) \\ \frac{r_{ma}}{r_a}(l_b \sin(\tilde{\theta}_{ma} + \tilde{\theta}_{mb}) + l_a \sin(\tilde{\theta}_{ma})) & \frac{r_{mb}}{r_b} l_b \sin(\tilde{\theta}_{ma} + \tilde{\theta}_{mb}) \end{bmatrix} \tag{15}$$

$$\tilde{\theta}_{ma} = -\frac{r_{ma}}{r_a}\theta_{ma} + \Delta\theta_a \tag{16}$$

$$\tilde{\theta}_{mb} = -\frac{r_{mb}}{r_b}\theta_{mb} + \Delta\theta_b \tag{17}$$

C. Device Capabilities

A number of different 2-DOF virtual environments have been implemented using Haplink. During rendering, we measured the maximum haptic loop rate at 500 Hz. Haplink is capable of delivering a range of maximum output forces between 2.7 to 14 N depending on its location in the workspace (Figure 4(b)) using the same Mabuchi motors as in the original Hapkit 3.0. In order to maintain a consistent feel in the virtual environments, we limit the maximum force throughout Haplink's workspace to 2.7 N in software. This workspace can be modified in shape and size as well as maximum force output at the end-effector by changing the device's kinematic parameters illustrated in Figure 5(b). Detailed performance quantification is left for future work.

V. Classroom Use

To date, Graphkit has been used in a 20-person introductory undergraduate robotics class taught by the last author in Kyoto, Japan to U.S. and Japanese students with a wide variety of majors (including Biology, Computer Science, Earth Sciences, Engineering, English, Math, and Physics). All students had previously taken an introductory programming course, and half the students had taken an electronics course.

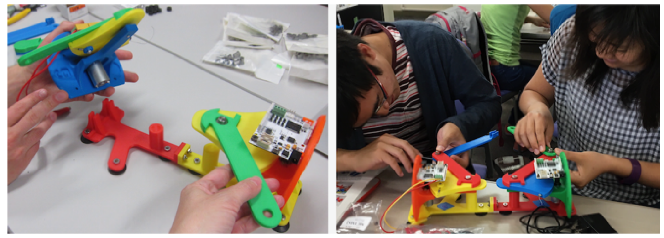


Fig. 7. Students building Graphkit in Kyoto, Japan.

Students began by assembling, analyzing, and programming Hapkit 3.0 both as an autonomous, simple, 1-DOF robot and as a 1-DOF haptic device. Then pairs of students assembled Graphkits and programmed the devices to function as 2-DOF drawing robots and 2-DOF haptic devices (rendering simple walls). All students in the course successfully completed the laboratories; with little noticeable difference in performance among students with engineering and non-engineering majors.

Figure 7 shows students in the class assembling and using their Hapkits and Graphkits. Lessons learned from this classroom implementation include the feasibility of using the Hapkit family of devices in a location remote from our research and development resources, as well as the role of basic programming knowledge as an equalizer for students with otherwise different coursework history. In future work, we will build on this initial application by quantitative and qualitative testing in K-12 educational environments.

VI. Discussion

One of the main challenges in designing 2-DOF haptic devices for educational use is ensuring that the devices can be assembled by students but are still robust enough to render high quality virtual environments. This was a fundamental requirement of the original Hapkit, driven by the idea that students learn by assembling the device themselves. However, maintaining this requirement for a higher-degree-of-freedom device becomes more challenging as the complexity of the device increases. With a larger number of interacting mechanical components, the effect of tolerancing is compounded. Grounding both motors for Haplink and Graphkit was key to this requirement – by reducing the inertia of the moving parts and thus the load on the structure, it allowed lower precision 3-D printed parts to be used.

Our goal in designing accessible 2-DOF open source devices is to increase the possibilities of what the research and education community can do with haptic devices and continue to foster education and haptics science. Hapkit has already had impact in advancing research and education. It has been used at Stanford University in undergraduate and graduate courses on haptics and controls, and it is being used in a self-paced massive open online class (MOOC) on haptics that has had over 5000 students enrolled. Hapkit is also being used in a collaboration between Stanford University and the University of British Columbia [35] [36] to understand the role of haptics in learning as well as how to best design course material that takes advantage of haptics. Users have also reported the role of Hapkit as a research tool; for example in human perception experiments and rat behavioral work. By using Hapkit 3.0 in different environments, we have learned that one of its best features is its capability to be customized and personalized by students, giving them a sense of ownership over their individual devices. This is why it is important to create customization processes that are accessible to K-12 students and educators who do not have access to expensive CAD tools.

As part of our future work we are developing a more advanced accessible customization process for more experienced users using FreeCAD. We are also working to further improve the design of our higher-degree-of-freedom devices as well as perform quantitative analysis of the devices' capabilities. One of our main challenges moving forward will be to create meaningful lessons using our higher-degree-of-freedom devices to integrate these devices into other courses, in order to further understand the capabilities of haptics in learning.

ACKNOWLEDGMENTS

The authors thank Prof. K. MacLean, O. Schneider, G. Minaker, Prof. P. Blikstein, and R. Davis for educational haptics collaborations. A. Lehto and A. Haas contributed to the Haplink design. T. Morimoto, A. Taylor, J. D. A. Pultorak, J. Wang, and A. Calasanz-Kaiser contributed to the original Hapkit 3.0 design.

REFERENCES

[1] The TAPR Open Hardware License Version 1.0. Accessed: 01.01.2017. [Online]. Available: http://www.tapr.org/ohl.html

[2] E. Rubow. (2008) Open Source Hardware: Technical Report. Accessed: 01.01.2017. [Online]. Available: https://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow_tr_report.pdf

[3] R. Stallman. (2015) Free Software, Free Society: Selected Essays of Richard M. Stallman. Accessed: 01.01.2017. [Online]. Available: http://www.gnu.org/doc/fsfs3-paperback.pdf

[4] R. Acosta, "Open Source Hardware," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.

[5] Arduino. Accessed: 03.16.2017. [Online]. Available: https://www.arduino.cc/

[6] RepRap. Accessed: 03.16.2017. [Online]. Available: http://reprap.org/

[7] OPENROBOTHARDWARE: An Initiative for Open Robot Hardware. Accessed: 03.16.2017. [Online]. Available: http://www.openrobothardware.org/

[8] R. Stallman. (1999) Richard Stallman- On "Free Hardware". Accessed: 01.01.2017. [Online]. Available: http://www.linuxtoday.com/infrastructure/1999062200505NWLF

[9] ——. (2015) Why we Need Free Digital Hardware Designs. Accessed: 01.01.2017. [Online]. Available: https://www.wired.com/2015/03/need-free-digital-hardware-designs/

[10] P. McNamara. (2007) Open Hardware. Accessed: 01.01.2017. [Online]. Available: http://timreview.ca/article/76

[11] M. Orta Martinez, T. K. Morimoto, A. T. Taylor, A. C. Barron, J. D. A. Pultorak, J. Wang, A. Calasanz-Kaiser, R. Lee Davis, P. Blikstein, and A. M. Okamura, "3-D Printed Haptic Devices for Educational Applications," in *IEEE Haptics Symposium*, 2016, pp. 126–133.

[12] C. Ramstein and V. Hayward, "The pantograph: A large workspace haptic device for a multi-modal human-computer interaction," in *CHI'94 Conference on Human Factors in Computing Systems*, 1994, pp. 57–58.

[13] C. Richard, A. M. Okamura, and M. R. Cutkosky, "Getting a feel for dynamics: Using haptic interface kits for teaching dynamics and controls," in *ASME IMECE 6th Annual Symposium on Haptic Interfaces*, 1997, pp. 15–21.

[14] D. I. Grow, L. N. Verner, and A. M. Okamura, "Educational haptics," in *AAAI 2007 Spring Symposium: Semantic Scientific Knowledge Integration*, 2007, pp. 53–58.

[15] K. Bowen and M. K. O'Malley, "Adaptation of haptic interfaces for a labview-based system dynamics course," in *IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2006, pp. 147–152.

[16] J. L. Gorlewicz and R. J. Webster III, "A formal assessment of the haptic paddle laboratories in teaching system dynamics," in *American Society for Engineering Education*, 2012.

[17] C. G. Rose, J. French, and M. K. O'Malley, "Design and characterization of a haptic paddle for dynamics education," in *IEEE Haptics Symposium*, 2014, pp. 265–270.

[18] C. E. Wong and A. M. Okamura, "The Snaptic Paddle: A Modular Haptic Device," in *IEEE World Haptics Conference*, 2005, pp. 537–538.

[19] A. Okamura, C. Richard, and M. Cutkosky, "Feeling is believing: Using a force-feedback joystick to teach dynamic systems," *Engineering Education*, vol. 91, no. 3, pp. 345–349, 2002.

[20] R. Gassert, J.-C. Metxger, K. Leuenberger, W. L. Popp, M. R. Tucker, B. Vigaru, R. Zimmermann, and O. Lambercy, "Physical student-robot interaction with the ETHZ haptic paddle," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 9–17, 2013.

[21] A. Otaran, O. Tokatli, and V. Patoglu, "Hands-On Learning with a Series Elastic Educational Robot," in *EuroHaptics Conference*, 2016, pp. 3–16.

[22] Hapkit: Open-hardware haptic device. Accessed: 01.01.2017. [Online]. Available: http://hapkit.stanford.edu

[23] G. Campion, Q. Wang, and V. Hayward, "The Pantograph Mk-II: A Haptic Instrument," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2005, pp. 723–728.

[24] C. Gallacher, A. Mohtat, S. Ding, and J. Kövecses, "Toward open-source portable haptic displays with visual-force-tactile feedback colocation," in *IEEE Haptics Symposium*, 2016, pp. 65–71.

[25] J. Forsslund, M. Yip, and E.-L. Sallnäs, "Woodenhaptics: A starting kit for crafting force-reflecting spatial haptic devices," in *Ninth international Conference on Tangible, Embeded, and Embodied Interaction*, 2015, pp. 133–140.

[26] R. B. Gillespie, M. B. Hoffman, and J. Freudenberg, "Haptic Interface for Hands-On Instruction in System Dynamics and Embedded Control," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 410–415.

[27] M. Shaver and K. Maclean, "The Twiddler: A haptic teaching tool for low-cost communication and mechanical design," Tech. Rep., 2003.

[28] B. Verplank, M. Gurevich, and M. Mathews, "THE PLANK: Designing a simple haptic controller," in *Conference on New Instruments for Musical Expression*, 2002, pp. 1–4.

[29] Tpad Tablet Project: A handheld haptic device with a variable friction surface. Accessed: 01.01.2017. [Online]. Available: http://tpadtablet.org/

[30] J. E. Colgate and J. M. Brown, "Factors affecting the Z-width of a haptic display," in *IEEE International Conference on Robotics and Automation*, 1994, pp. 3205–3210.

[31] M. K. O'Malley and M. Goldfarb, "The implications of surface stiffness for size identification and perceived surface hardness in haptic interfaces," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 1255–1260.

[32] R. Ellis, O. Ismaeil, and M. Lipsett, "Design and evaluation of a high-performance haptic interface," *Robotica*, vol. 14, no. 3, pp. 321–327, 1996.

[33] V. Hayward and O. R. Astley, "Performance measures for haptic interfaces," in *Robotics Research*, 1996, pp. 195–206.

[34] M. Kilchenman and M. Goldfarb, "Force saturation, system bandwidth, information transfer, and surface quality in haptic interfaces," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 1382–1387.

[35] G. Minaker, O. Schneider, R. Davis, and K. E. MacLean, "HandsOn: Enabling Embodied, Creative STEM e-learning with Programming-Free Force Feedback," in *EuroHaptics Conference*, 2016, pp. 427–437.

[36] R. L. Davis, M. Orta Martinez, O. Schneider, K. MacLean, A. Okamura, and P. Blikstein, "The haptic bridge: Towards a theory of haptic-supported learning," in *Proceedings of the 16th International Conference on Interaction Design and Children*, ser. IDC '17. ACM, 2017.