SURVEY

Open Robot Hardware

Progress, Benefits, Challenges, and Best Practices

By Vatsal V. Patel, Minas V. Liarokapis, and Aaron M. Dollar


©SHUTTERSTOCK.COM/IURII MOTOV

Technologies from open source projects have seen widespread adoption in robotics in recent years. The rapid pace of progress in robotics is in part fueled by open source projects, providing researchers with resources, tools, and devices to implement novel ideas and approaches quickly. Open source hardware, in particular, lowers the barrier of entry to new technologies and can further accelerate innovation in robotics. But open hardware is also more difficult to propagate in comparison to open software because it involves replicating physical components, which requires users to have sufficient familiarity and access to fabrication equipment. In this work, we present a review on open robot hardware (ORH) by first highlighting the key benefits and challenges encountered by users and developers of ORH, and then relaying some best practices that can be adopted in developing successful ORH. To accomplish this, we surveyed more than 80 major ORH projects and initiatives across different domains within robotics. Finally, we identify strategies exemplified by the surveyed projects to further detail the development process, and guide developers through the design, documentation, and dissemination stages of an ORH project.

BACKGROUND

The open source paradigm of disseminating new technologies has gained traction over the last several decades, particularly in rapidly evolving deep tech fields such as robotics, for its benefits in expediting innovation through the sharing of knowledge, tools, resources, and technical solutions with a community of researchers, citizens scientists, hobbyists, and technology enthusiasts. The rapid pace of progress in robotics has been due in part to the availability of high-quality open source software and hardware solutions that allow roboticists to easily use, adapt, and improve them in their own applications, subsequently accelerating the implementation and testing of novel ideas in a variety of robotics domains. Particularly, open source software has seen widespread proliferation in robotics [1], [2] in the form of middleware suites [3], [4], computer vision libraries [5], robot control packages [6], simulation environments [7], [8], and robot motion planning platforms [9] among others.

In comparison, open source hardware projects have trailed software in the number and type of hardware available [10]. In general, hardware is more challenging to propagate than

Digital Object Identifier 10.1109/MRA.2022.3225725
Date of current version: 19 December 2022

software because of the difficulties encountered when replicating physical components as opposed to using and modifying code. Researchers often choose to purchase commercially available, expensive, closed source hardware that is hard to modify, repair, and maintain, or dedicate resources and time to creating their own hardware from the ground up. But recent advancements in rapid prototyping (e.g., 3D printing and laser or waterjet cutting) have enabled mechanical fabrication with minimal specialized expertise, easy-to-use equipment, and at relatively low cost, especially compared to alternatives such as computerized numeric control (CNC) machining. These technologies, although still in their infancy, have made it so that many parts can be fabricated with sufficient durability and mechanical strength to be functionally used in a range of robotics applications. As a result, building an open source mechanical hardware project has become much more feasible, and there are now a range of high-quality open source hardware projects that create new technologies and deliver impact to the robotics community. The objective of this review is to highlight such robot hardware by identifying the key characteristics and effective development practices, surveying widely adopted projects in various robotics domains, and organize an open source hardware development process with best practices and strategies derived from the surveyed projects.

Few previous reviews of open source hardware have been conducted, and to the authors' knowledge, none has taken an all-encompassing look at hardware across the various fields within robotics. Reviews have looked at open source projects in specific fields of robotics, such as the design of medical devices [11], unmanned aerial vehicles (UAVs) [12], hardware in research labs [13], and in educational applications [14], [15]. Initiatives such as the ORH website [16] and the Open Source Hardware Association's (OSHWA's) directory host a compilation of open source mechanical and electrical hardware [17]. In addition to its directory of projects, OSHWA also promotes open source hardware use to general audiences, establishes shared principles of the open source hardware movement, and certifies projects that meet its standard of open source compliance. More recently, academic journals dedicated to open source design in science [18], [19], [20] and special issues [21] have recognized the impact of open source hardware specifically in robotics.

The remainder of this article is organized as follows. The "ORH" section focuses on what constitutes ORH as well as on the characteristics of such hardware, noting the advantages and challenges it brings to both users and developers, along with best practices for developers to create effective ORH projects. The "Survey of ORH Projects" section presents a survey of more than 80 major ORH projects categorized by the robotics subfield they are applied within. Finally, the "Developing an ORH Project" section describes the different stages in the process of developing new ORH projects, highlighting the strategies adopted by successful ORH projects. The characteristics, best practices, and surveys from this article will also be made available through the ORH website [16].

WHAT IS OPEN ROBOT HARDWARE

In this work, we review projects that can be ascribed as ORH, which we define as satisfying the following inclusion criteria:

- *Open source*: The design and auxiliary part files required for redesign, fabrication, and assembly are made public, along with the documentation that facilitates replication of the work. The work should be disseminated with a license that allows use, replication, modification, and sharing of the various project components under the license's terms and conditions.
- *Robotics oriented*: The project has applications in a robotics domain, either as a complete robot system or constituent of a robot system.
- *Mechanical hardware oriented*: The project focuses on physical items with tangible mechanical components, such as structural framework, mechanisms, actuators, and sensors. The projects with supplementary electronics or software elements that support the hardware are included, but projects with solely electronics or software elements are excluded from the scope of this review.

We also define the user of an ORH in the context of this review as an entity replicating, using, modifying, or even simply drawing inspiration from an ORH project. The user is then at the receiving end of the content output by the ORH's developer, defined here as the designer, distributor, author, or any entity involved in the development, design, documentation, or dissemination of the project. It is possible

TABLE 1. The characteristics of ORH.

	USER	DEVELOPER
Advantages	• easy to use, modify, and repurpose • lower cost to acquire and service • easy to repair and maintain • upgradable on site and in the future • access to the user community for large projects.	• decentralized co-development and covalidation • gains valuable and quick user feedback • new applications of technology explored • engages a large user community • exposure by delivering impact to community.
Challenges	• requires appropriate tools, equipment, and skills • requires time and labor for replication • lacks guaranteed, long-term support • lacks comprehensive documentation • lacks guaranteed reliability or robustness	• prepares comprehensive documentation • guarantees design and fabrication simplicity • keeps cost low for affordability • selects license and dissemination strategy • nurtures the engaged user community • captures value from hardware

for the developer to also take on the role of the user if he or she is redesigning or adapting an existing open source project. It is also possible for a user to become a part of the development team, which can occur when the community actively engages with the developers of popular open source hardware projects.

CHARACTERISTICS OF ORH

To understand why someone would consider developing or using an ORH project, it is important to delve into the advantages and challenges associated with such hardware from the perspective of both the users and the developers separately (summarized in Table 1 and also available on the ORH website [16]).

ADVANTAGES FOR THE USER

One of the primary goals of open source projects is to allow a community of users to build on a specific work, sharing knowledge, know-how, resources, tools, technical solutions, and documenting all the issues and problems encountered during the development process. The key advantage that an ORH project offers over commercially available alternatives is customization. Users have the flexibility to understand how the system works, modify it, and improve it to meet their requirements or the specifications for a particular application. Particularly in a rapidly growing field such as robotics, where new applications are regularly identified, hardware adaptability can significantly help condense the development timeline. Although creating their own hardware from scratch might offer users even more flexibility, the time and human labor allocated to its development could prove to be prohibitive. Commercial robot hardware products, on the other hand, are significantly costlier to purchase or license in comparison to ORH, with additional servicing fees for repairs and upgrades. An ORH project thus presents a more economical option for the user to quickly test whether the hardware is appropriate for his or her implementation without any long-term commitment.

“PARTICULARLY IN A RAPIDLY GROWING FIELD SUCH AS ROBOTICS, WHERE NEW APPLICATIONS ARE REGULARLY IDENTIFIED, HARDWARE ADAPTABILITY CAN SIGNIFICANTLY HELP CONDENSE THE DEVELOPMENT TIMELINE.”

Another advantage ORH offers users over commercial products is repairability. A purchased product might either require a service person visit or a return to the manufacturer, a limitation magnified in hardware components in comparison to software, which could be patched remotely. Similarly, commercial hardware often does not allow users to perform maintenance on their own. On the other hand, the ORH could be repaired and maintained on site by the user. Although this might require sufficient knowledge of the hardware components, popular ORH projects further benefit from their strong user community and developer input, which can offer quick support for common repairs and hardware fixes. In this regard, ORH can notably reduce downtime and expense of restoring the robot or robot component back to function.

The swift pace of development in robotics often renders components of the robot unusable and out of date within a few years. Robot software developers may be able to remotely ship updates on their products to maintain compatibility and patch issues, whereas robot hardware manufacturers often require users to purchase a new version of their product. In contrast, ORH allows the user to decide if and when they want to upgrade. ORH users can also choose to only upgrade specific components or functions of the hardware. In commonly used ORH, the active user community as well as the developer regularly release these updates to the hardware through design changes or extending alternative component options.

CHALLENGES FOR THE USER

Many ORH projects require the user to fabricate and assemble parts on his or her own, possibly while using unfamiliar tools and processes, such as molding with 3D-printed parts for hybrid deposition manufacturing (HDM) [22]. The knowledge barrier of reproducing hardware can be more pronounced for some users, especially when replication of the hardware relies on sophisticated equipment, resources, and tools, or on significant skills and expertise of users. ORH developers may release bad documentation and insufficient fabrication and assembly instructions, which further raise the barrier for the nonexpert user. This lack of subject-area expertise is one of the key hurdles ORH users face in the early stages of hardware implementation. Even if users have the expertise and access to requisite manufacturing tools, they may not be able to allocate the time required to fabricate and assemble the components. Some developers will sell prefabricated parts or even completely preassembled versions of their hardware to bypass the user knowledge and time required to put their hardware together. Well-funded research groups typically choose the route of buying the part kits, whereas other teams would try to replicate the parts themselves using any available resources. But even with these purchased ORH kits and components, the user retains the advantages of ORH, such as repairability, upgradability, and adaptability outlined in the previous section, while circumventing the challenges of recreating the hardware on his or her own.

Once an ORH project is developed, tested, and disseminated, users create small or large communities that inform the core development team about issues that may arise and also contribute to the further development or modification of the hardware to suit different needs. Unlike commercially available hardware that might offer service contracts and maintenance assistance, an ORH component can suffer from inadequate long-term support if the community becomes inactive over time, or even a loss of access to the original files if the developer takes down the project website or stops maintaining the project (e.g., the CAD files may no longer be supported). Although ORH enables the user to repair the hardware on site as mentioned previously, the guidance required to perform that repair may be lacking without a formal support team setup to help debug and tackle potential failures over long-term use of the ORH. In fact, components of the ORH might not be suitable for extended use, in comparison to commercially available products that come with a warranty and would have been tested beforehand to meet a certain number of hours in operation, and to acquire particular certifications. The upside of reliability and robustness from commercial hardware is often compensated for by the repairability of ORH, but still, the long-term support and reliability of ORH could be another challenge for users.

> THE TRANSPARENT NATURE OF ORH IMPROVES THE HARDWARE BY EVALUATING IT MORE RIGOROUSLY AND CO-DEVELOPING IT WITH NUMEROUS USERS.

ADVANTAGES FOR THE DEVELOPER

A robot hardware developer needs to decide whether he or she envisions his or her end product to be open sourced so that design and fabrication choices are compatible with the design release for even a novice user to implement. Some of the main benefits motivating developers to open source their hardware is that their technology reaches more people, delivers more impact to the community, attracts more citations to their work, and establishes the developer as a leader in the field. To achieve this, developers should aim to design their hardware with reproducibility in mind, such as by incorporating rapid-prototyping techniques and off-the-shelf components. The ORH is more likely to spread through the community in this manner, and more users may adopt the hardware to use in their own projects, or modify and release their own derivatives. Users may also demonstrate new promising applications for the hardware in other areas of robotics. So the more widely adopted an ORH becomes, the more exposure it generates for the developer, established through the vibrant user community. Thus, this can present the developer with new opportunities for collaborations, such as through updates and patches recommended by users, and create a base for the developer to release more ORH in the future.

As the design, fabrication, and auxiliary files of an ORH are all released to the public, the hardware and its function are more thoroughly validated through implementation by a number of users. This effectively decentralizes the R&D process of the hardware among the user community. In comparison, for instance, the developer could run only a few experiments to test every component on his or her own, whereas open sourcing the hardware practically recruits lots of co-testers and co-developers quickly, who would utilize the hardware in their own disparate application areas and notify the developer about issues with the hardware that might not have been foreseen. The transparent nature of ORH improves the hardware by evaluating it more rigorously and co-developing it with numerous users. Decentralization of the R&D is also much faster and less expensive than the R&D conducted only by the developer. However, this does not mean that the developer may release incomplete or poorly designed hardware, expecting that user community will identify all the problems with the hardware, because in such a case, the technology will not be adopted at all and the community will never be formed.

CHALLENGES FOR THE DEVELOPER

A good open source project should be accompanied by comprehensive documentation and detailed user guides that can help the user replicate, use, repair, and maintain the hardware. Writing such in-depth documentation and directions requires the developer to devote substantial time and effort and can be particularly challenging if the developer lacks prior experience in doing so. Although the developer can simply release just the design files without any guides or details, such an ORH project is much less likely to be picked up and widely used by the user community. So to recover the benefits of an ORH, as outlined previously, the developer must also dedicate resources and time to prepare a well-documented ORH. It should also be noted that such documentation is not required in closed source projects, which are typically accompanied by very basic user guides that cover only operation of the device.

Although supporting documentation can be written toward the end of the development process, the developer should consider open sourcing the project early on to incorporate simplicity into the design and the required fabrication processes and make the released ORH accessible to as many users as possible. Evaluating the complexity of the hardware early in development can prevent the developer from having to redesign components and modify features to be compatible with common fabrication processes. Simplicity of design also extends to avoiding lots of custom parts, which could be replaced with readily available alternatives. The developer needs to actively consider how his or her design choices, fabrication processes, and component selection affect the accessibility of the ORH. And doing so early on can temper some of these challenges in creating hardware that gets implemented by a large number of users.

In addition to documenting the hardware and including easily sourced components, the developer should also assess the total cost of building and operating his or her ORH. One of the main advantages of using open hardware is the lower cost compared to purchasing commercial products, which may come with licensing or service fees on top. However, if the combined cost of components in the ORH, along with the time and effort required to manufacture and assemble parts, reaches or even exceeds the cost of the commercially available hardware alternative, users will be less likely to adopt the ORH. Keeping the total price of off-the-shelf parts, calling for simple fabrication processes, and requiring common materials and tools for assembly can aid affordability of the ORH and improve its ability to reach new users.

Once the hardware has been designed and documented, developers face the challenge of selecting a dissemination strategy and an appropriate license that will help them reach as many potential users as possible. Even after users adopt the hardware, developers need to continue engaging and nurturing the community around the project. This might require them to regularly communicate through forums, incorporate user feedback, update files for compatibility, or prepare additional documentation. For all the time and labor required, prospective developers considering open sourcing their project may be put off by the unclear opportunities to capture value for themselves through the released hardware. There is not a straightforward commercialization strategy for open source hardware, which may be a significant challenge, especially for developers with limited funding to continue R&D on their project. In such cases, developers could choose to apply certain licenses to capitalize on any commercial distribution of their hardware in the future.

BEST PRACTICES FOR ORH

A successful ORH project maximally leverages developers' efforts in preparing documentation, maintaining design simplicity and budget, and publishing regular updates by extracting all the benefits that come with open source technologies. As such, the developer should try to gain widespread adoption of his or her project and sustain it over a long period of time. OSHWA lays out steps developers can take to persuade users to use their open source project and help build a community around it [17]. In this section, we briefly outline some of these best practices relevant to ORH that developers could follow after they have decided to open source their robot hardware (see Figure 1). Later, in the "Developing an ORH Project" section, we further detail the pipeline of requisite methods and considerations at these different stages of developing an ORH project from scratch, along with examples from effective ORH projects.

- The ORH development process begins with the design of the hardware components. As an open source hardware project, the original and editable—preferably, made with open source CAD tools like FreeCAD or Onshape—design files for the various custom components in the assembly should be released for the user to reference and fabricate the parts. This includes 3D CAD files and 2D part files for planar parts, in various editable and interchange formats like .step.
- These design files should be prepared with ranges of user expertise in mind, i.e., the design features should be purposefully simple and consistent throughout the parts. In addition to part files, supplementary files like circuit diagrams and technical drawings may also be useful for the user to replicate the components easily on his or her end.
- Off-the-shelf components should be utilized as much as possible to relieve some of the manufacturing load of replicating an ORH. If custom-fabricated parts are absolutely required, then the design of these parts should account for easy manufacturability at the user's end and the variability in fabrication from different types of manufacturing equipment.
- Although the source part files are a crucial part of an ORH project, they are not sufficient on their own to ensure the hardware's reproducibility. Thorough documentation describing the various manufacturing and assembly steps needed to build the hardware makes open source projects extremely lucrative. The bill of materials (BOM) and lists with links to off-the-shelf parts are very useful for users to acquire components.
- Before prospective users commit to an ORH, they might also want to know whether they have access to the required equipment and whether they can use the fabrication processes needed to replicate all the components of the project. The documentation should list fabrication

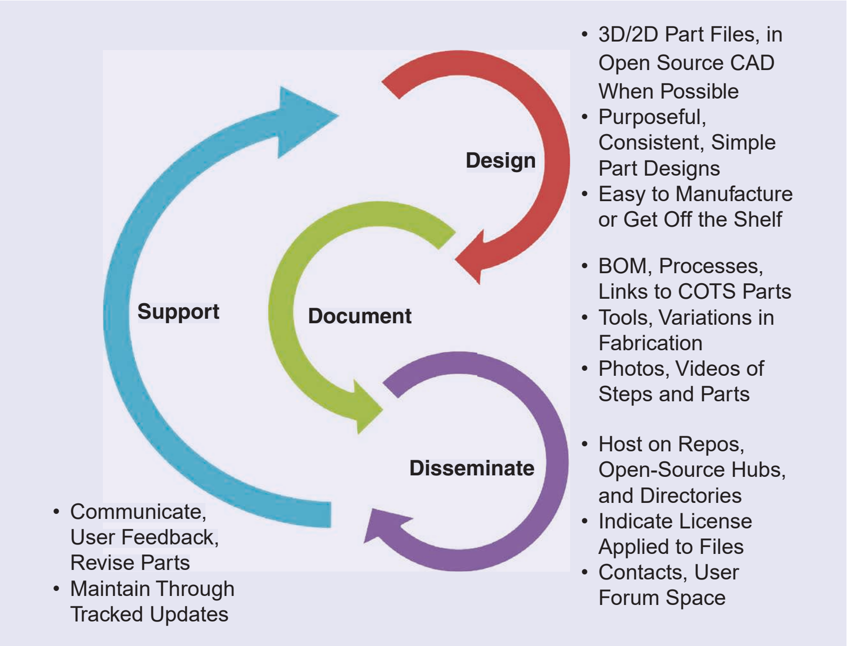

FIGURE 1. The development process of an ORH with some best practices that developers can follow to make their project as widely adopted and retained as possible. BOM: bill of materials; COTS: commercial off the shelf.

methods next to the components as well as any tools needed for postfabrication and assembly, allowing for alternate methods and equipment where possible.

- Although more information for the user is better than less, a good documentation package ensures that the instructions are presented in a logical and clear manner. To this effect, annotated images or videos can go a long way in communicating complex and multistep building processes effectively to even the novice user.
- Once the design files and documentation packages have been prepared, developers can disseminate the project through online repositories or project websites to reach as many users as possible. There are also several open source project hubs (such as OSHWA and ORH directories).
- For disseminating within academic communities, journals dedicated to open source hardware can further spread the reach of the ORH. Sending e-mails on popular research mailing lists, publishing research papers, and participating in conferences and workshops can also help increase the user base.
- Indicating the license applied to the ORH files prominently on the site can inform prospective users early on and prevent any breach of the user agreement.
- Developers can also support and nurture their user community by creating forum spaces for users to interact with each other as well as provide feedback on the project. An active user base for the ORH amplifies the aforementioned advantages of open sourcing the project for the developer.
- With the pace of progress in rapidly growing fields like robotics, it is crucial to update and revise the ORH files regularly to support current users. Updates should be tracked and released to add compatibility for new versions of off-the-shelf parts, or to account for novel fabrication methods. Users can inform the developer when and which updates may be needed to the project, and even prepare these revisions themselves if the project files are modifiable in freely available CAD software.

The development of a successful ORH is thus continually iterative: using the feedback from support provided to users in informing design changes, which subsequently gets documented and disseminated on the platform in a process that gets repeated through the lifetime of the ORH. Thus, an ORH project can be made vastly more effective by employing these best practices at different stages of the process. In the "Developing an ORH Project" section, we further detail these stages of the development process, focusing on the tangible steps prospective developers can take when building their ORH.

"THE DEVELOPMENT OF A SUCCESSFUL ORH IS THUS CONTINUALLY ITERATIVE: USING THE FEEDBACK FROM SUPPORT PROVIDED TO USERS IN INFORMING DESIGN CHANGES, WHICH SUBSEQUENTLY GET DOCUMENTED AND DISSEMINATED ON THE PLATFORM IN A PROCESS THAT GETS REPEATED THROUGH THE LIFETIME OF THE ORH."

SURVEY OF ORH PROJECTS

The development and dissemination of open hardware can vary depending on its scope and purpose within the application area. As robotics encompasses a wide range of fields, each requiring unique hardware and using distinct design and fabrication methods, the primary attribute used for the categorization of ORH in this article is that of the hardware's application domain within robotics. The list of ORH projects classified here is certainly not exhaustive, but rather aims to identify the practices of some of those projects that led to their widespread adoption within the user community.

The surveyed projects for which the files are still accessible are listed in Table 2 and also published through the ORH website [16]. In this table, we note the ORH properties and practices, followed by each project so that prospective developers can identify the current ORH with the desired attributes.

ROBOT ARMS AND HANDS

Applications in grasping and manipulation require the robot to be outfitted with different end effectors depending on the task and objects used. The choice of end effector is often limited by compatibility with the robot arm. Open source robot hands seek to expand the robot systems' functionality by offering inexpensive and customizable robot hands that users can implement for their own use cases. The Yale OpenHand project [23] sought to create a library of such low-cost, 3D-printed hand designs for researchers to use [see Figure 2(a)]. The source design files are available to public, along with a fabrication guide and videos to aid users in reproducing the hands on their end [27]. The Baxter EasyHand [28] draws from the OpenHand designs to offer an even less expensive and easy-to-build robot hand that functions with existing actuators on the Baxter robot arm. The EasyHand project exemplifies an instance of a derivative ORH project modifying and redistributing another ORH project. Similarly, the OpenBionics initiative [29], initially inspired by the OpenHand project, releases several open source robot and prosthetic hands, such as a series of modular robot hands [30]. Open robotic hand platforms also present users with an opportunity to standardize evaluations of algorithms through benchmarking test setups. Several such platforms have been developed for testing learning algorithms for dexterous manipulation, such as the ROBEL D'Claw [31] and TriFinger [24] platforms [see Figure 2(b)]. Both of these three-fingered hands aim to lower the cost barrier for experimentation with a full-commercial robot system and are intended to be used on standalone mounts without a robot arm.

Anthropomorphic hands form a large portion of robot hand designs used to carry out grasping and manipulation tasks in human spaces. The human–robot interaction (HRI) hand [32] presents an anthropomorphic end effector as a research platform for collaborative robotics that can be built for a lower cost than acquiring a commercial gripper. A derivative version of the adaptive anthropomorphic hand of OpenBionics [33] aims to be capable of carrying out more dexterous manipulation tasks with fingers that can also move in the abduction/adduction direction. Human-like robot hands are also pivotal to prosthesis hand development. Such hand designs are often open sourced for affordability compared to commercially available prostheses and provide a research platform for evaluating control algorithms, human–machine interfaces, and operation schemes in powered prosthetics. To achieve these objectives, open prosthetic hand designs maintain a low complexity of design so that the hands are easily reproducible. The OpenBionics prosthetic hand [34] uses just a single actuator with a lockable differential mechanism to permit locking different fingers in position. Anthropomorphic hands such as the Open Source Hand [35] can also be created with the aim of serving as a testbed to implement different prosthetic control strategies on a six-degree-of-freedom (6-DoF) hand. The PUT hand [36] combines fully actuated and underactuated fingers in its anthropomorphic design, with the goal of manipulating elastic objects. Similarly, the Brunel hand [37] can be used as a platform for researchers in prosthetics, grasping, manipulation, and HRI. This hand is available for purchase from Open Bionics Labs but can also be fabricated using the design files provided by the developer.

Similar to prostheses, open source exoskeletons for arms and hands serve to increase accessibility to wearable, assistive devices for those who have lost some upper-limb functionality. The low-cost ExoArm v2 [38] exoskeleton aims to assist elderly or disabled individuals with lifting objects and with rehabilitation of arm motion. Exoskeletons for hands or exogloves can restore grasping capabilities and finger mobility in the hand [see Figure 2(c)]. The exoglove in [39] can be either body powered or motor driven, and Sparthan [40] features 3D-printed ring structures. These type of hand exoskeletons are often tendon driven, but some have also added pneumatic actuation for modulating stiffness and inflating a telescoping thumb [25]. Linkage-based actuation is also common in hand braces and orthoses, such as the purely passive wrist-driven orthosis that was open sourced in [41] and inspired the adapted motor-driven version [42]. Beyond assistive devices, exogloves also have a lot of applications in virtual reality and haptics. In [43] and [44], finger motions and pinch gestures are used to manipulate objects in virtual reality and simulations. Exogolves can be used to transmit haptic feedback to the wearer triggered by a variety of sensors, such as underwater ultrasonic range data indicated by micropumps varying pressure at the fingertips [45], or spatial position observed through a camera translated to vibration stimuli [46].

In comparison to robot hands, prostheses, and exogloves, open source multi-DoF robot arms are less commonly shared, likely due to the size and complexity of the whole arm design. Larger arms capable of carrying heavier payloads would also need components to be fabricated, using metal machining or other manufacturing processes that may not be accessible to many users. The Niryo One [26] [see Figure 2(d)], Thor robot arm [47], and CM6-compliant arm [48] with six DoF attempt to bridge the fabrication gap by requiring mostly 3D-printed components. And other robot arms such as the OpenManipulator-X [49] are made entirely from off-the-shelf components from ROBOTIS. Neither of these robot arms have a payload of more than 500–750 g. In comparison, the 3-DoF printed articulated robotic arm (PARA) [50] and 5-DoF Dexter [51] robot arms have higher payloads of 1–3 kg and use acrylic tubes and carbon-fiber plates for structure, respectively. Still, larger 6- or 7-DoF robot arms are particularly absent from the open hardware space. Although users might not be able to fabricate components for heavier payload arms, open sourcing the design files and releasing documentation still has the benefits of allowing users to repair, modify, and upgrade the hardware, as outlined previously in the "Characteristics of ORH" section.

SOCIAL ROBOTS

HRI is an active area of research in robotics, and social robot platforms are a central apparatus that researchers have used to test expressive gestures and behaviors in real-world experimental settings. Commercially available products may not be suitable to the type of empirical evidence researchers hope to collect, and this could often require the researchers to develop or adapt their own hardware. The MyKeepon project [52] [see Figure 3(a)] modified an off-the-shelf toy product and released a programmable, low-cost platform that could be built and repaired by users, allowing even multiple instances of MyKeepon robots to be deployed for comparative studies. On the other hand, some HRI experiments may call for customized and unique social robots.

The Open Source Social Robot Platform (OPSORO) [56] allows the user to produce novel embodiments of social robots from a set of modules atop a skeletal frame for an accelerated design cycle. Users can incorporate different modules of the platform to enact facial features and design distinct outer layers, allowing for a wide variety of robots to be created with the platform. In a similar vein, the Blossom social robot [53] can be custom built with novel handcrafted exteriors and outfitted with a range of crafted parts such that no two robots have to look alike [see Figure 3(b)]. Both the internal and external structures of Blossom are fabricated from soft materials to add compliance in the natural motions of the robot. Soft actuators and compliant mechanisms are also employed in the CASTOR social robot [57]. CASTOR has a partially humanoid-like appearance and is intended for use in therapy for children with autism spectrum disorder. Humanoids deployed as social robots and beyond are discussed in the next section in more detail. All of these open social robot platforms can be constructed using inexpensive materials and accessible components, and associated control software and assembly instructions are made available for users to implement the robots easily, such as the well-organized wiki for CASTOR [58]. Social robots thus represent a model section of the wider ORH community, where commercially available products may not appropriately match the type of physical

TABLE 2. ORH-surveyed projects.

PROJECT	DESIGN AND FABRICATION: EDITABLE CAD	CAD SW	FABRICATION METHODS	CAN IT BE PURCHASED?	COST	DOCUMENTATION AND INSTRUCTION: FABRICATION INSTRUCTIONS	ASSEMBLY GUIDE	BOM/COTS LINKS	ANNOTATED IMAGES	VIDEOS/ ANIMATIONS	TOOLS REQUIRED	DISSEMINATION AND SUPPORT: ONLINE REPOSITORY	OS LICENSE	SOFTWARE/ CONTROL	COMMUNITY FORUM	UPDATES SINCE 2020
Robot Arms and Hands																
OpenHand [23]	✓	SolidWorks	3D printing, HDM		US$150–200 plus motors	✓	✓	✓	✓	✓	✓	GitHub, website	CC BY-NC 3.0	✓		✓
OpenBionics [29]	✓	SolidWorks, FreeCAD	Laser cutting, HDM, 3D printing, stitching		US$100–200 plus motors	✓	✓	✓	✓	✓	✓	GitHub, website	CC BY-SA 4.0	✓		✓
ROBEL [31]	✓	SolidWorks	3D printing, laser cutting	✓	US$3,500	N/A	✓	✓	✓			GitHub, website	Apache 2.0	✓		
TriFinger [24]		Unknown	3D printing, machining		US$5,000	✓	✓	✓	✓		✓	GitHub, website	BSD 3-Clause	✓	✓	✓
HRI hand [32]	✓	Fusion 360	3D printing		US$500–900		✓	✓	✓			OSF, GitHub	MIT License	✓		
Open Source Hand [35]	✓	SolidWorks	3D printing		US$2,500							Website	Unknown			
PUT hand [36]	✓	Inventor	3D printing, machining		Unknown							GitHub	CC BY-NC 4.0	✓		
Brunel hand [37]		Unknown	3D printing	✓	US$1,500							GitHub, website	CC BY-SA 4.0	✓		
ExoArm v2 [38]		Unknown	3D printing		Unknown			✓				Hackaday, GitHub	Unknown	✓		
WDO [41]	✓	SolidWorks	3D printing		US$15	✓	✓	✓	✓		✓	Pinshape	CC BY-NC 4.0			
OMG VR gloves [43]		Unknown	Sewing, 3D printing		US$50–100	✓	✓	✓	✓	✓	✓	Instructables	CC BY-SA 4.0			
Lucid VR gloves [44]		Unknown	3D printing		US$60	✓	✓	✓		✓	✓	Hackaday, GitHub	MIT License	✓	✓	✓
IrukaTact [45]	✓	TinkerCAD	3D printing		Unknown			✓				Website	Unknown	✓		
Unfolding space glove [46]	✓	STEP	Sewing		US$500	✓	✓	✓	✓			Website, GitHub, Hackaday	CC BY 4.0	✓		✓
Thor arm [47]	✓	FreeCAD	3D printing		US$400	✓	✓	✓	✓	✓	✓	Website, GitHub, Hackaday	CC BY-SA 4.0	✓	✓	✓

CM6 arm [48]	✓	STEP	3D printing		US$1,000		✓	✓	✓			Hackaday, GitHub	MIT License	✓		
OpenManipulator-X [49]	✓	Onshape	3D printing	✓	US$1,500		✓	✓	✓	✓		Website, Thingiverse	CC BY 4.0	✓		✓
PARA arm [50]	✓	SolidWorks	3D printing		US$3,400	✓	✓	✓	✓		✓	OSF, GitHub	MIT License	✓		
Niryo One [26]		Unknown	3D printing	✓	US$1,600		✓	✓	✓	✓	✓	GitHub, website	CC BY-NC-SA 4.0	✓		
Dexter arm [51]	✓	Onshape	3D printing	✓	$3,000		✓	✓		✓		GitHub, Hackaday	GPL v3	✓	✓	✓
Social Robots																
OPSORO [56]	✓	Rhino3D	3D printing, laser cutting		Unknown		✓	✓	✓		✓	Website, GitHub	MIT License	✓		
Blossom [53]	✓	2D, SolidWorks	Laser cutting, knitting, 3D printing		US$250	✓	✓	✓	✓		✓	GitHub	MIT License	✓		✓
CASTOR [57]		Unknown	3D printing, sewing		Unknown		✓	✓	✓	✓		GitHub	CC BY-NC-SA 4.0	✓		✓
Humanoids																
DARwin-OP, ROBOTIS-OP2 [59]	✓	SolidWorks	CNC mill, band saw, Dremel, bending break	✓	US$9,600	✓	✓	✓	✓		✓	GitHub, Source-Forge	Apache 2.0	✓		
ROBOTIS-OP3	✓	STEP	3D Print	✓	$12,000							GitHub	Apache 2.0	✓		
NimbRo-OP2X [61]	✓	STEP, IGES	3D printing, machining		Unknown			✓				GitHub	CC BY-NC-SA 3.0	✓		
Poppy [54]	✓	SolidWorks	3D printing		US$9,000		✓	✓	✓	✓	✓	Website, GitHub	CC BY-SA 4.0	✓	✓	✓
InMoov [62]		Unknown	3D printing	✓	US$1,500	✓	✓	✓	✓	✓		Website	CC BY-NC 3.0	✓	✓	✓
Epi [65]	✓	STEP	3D printing		US$8,500				✓			GitHub	CC BY-SA 4.0	✓		
NICO [66]	✓	Inventor	3D printing, laser cutting	✓	Unknown		✓	✓	✓	✓		GitHub, website	CC BY-SA 3.0	✓		
Eva [67]	✓	SolidWorks	3D printing, laser cutting, molding		US$900	✓	✓	✓	✓			Zenodo	BSD 3-Clause	✓		
iCub [55]	✓	PTC Creo	Machining, molding	✓	US$260,000		✓	✓	✓	✓		Website, GitHub	BSD 3-Clause, GPL v2	✓	✓	✓

(Continued)

TABLE 2. ORH-surveyed projects.(*Continued*)

PROJECT	DESIGN AND FABRICATION					DOCUMENTATION AND INSTRUCTION					DISSEMINATION AND SUPPORT					
	EDITABLE CAD	CAD SW	FABRICATION METHODS	CAN IT BE PURCHASED?	COST	FABRICATION INSTRUCTIONS	ASSEMBLY GUIDE	BOM/COTS LISSNKS	ANNOTATED IMAGES	VIDEOS/ANIMATIONS	TOOLS REQUIRED	ONLINE REPOSITORY	OS LICENSE	SOFTWARE/CONTROL	COMMUNITY FORUM	UPDATES SINCE 2020
Legged Robots																
Open Dynamic Robot Initiative [69]		Unknown	3D printing		US$4,200	✓	✓	✓	✓		✓	GitHub, website	BSD 3-Clause	✓	✓	✓
Doggo [75]	✓	Fusion 360	Waterjet cutting, 3D printing, machining, routing		US$3,000		✓	✓	✓			GitHub	MIT License	✓	✓	
Pupper [76]	✓	Fusion 360	3D printing, routing	✓	US$900		✓	✓	✓	✓	✓	ReadThe Docs	MIT License	✓	✓	
SpotMicro [77]		Unknown	3D printing		Unknown		✓	✓	✓	✓	✓	Thingiverse, ReadThe Docs	CC BY 4.0	✓	✓	
OpenQuad. [78]	✓	STEP	3D printing		US$600		✓	✓				GitHub	MIT License	✓		
Oncilla [70]	✓	SolidWorks	3D printing		Unknown							c4science	GPL v3	✓		
Metabot [80]	✓	OpenSCAD	3D printing, laser cutting	✓	US$400			✓	✓			GitHub	CC BY-NC 3.0	✓		
Aracna [81]	✓	SolidWorks	3D printing		US$1,400			✓				GitHub	GPL v3	✓		
DyRET [82]		Broken Link	3D printing		US$6,500							GitHub	CC BY-SA 4.0	✓		
OpenRoACH [83]	✓	SolidWorks	3D printing, laser cutting		US$150	✓	✓	✓	✓			Website, Google Drive	Unknown	✓		
HOPPY [71]	✓	SolidWorks	N/A		US$500	N/A	✓	✓		✓		GitHub	Unknown	✓		
Open Source Leg [72]	✓	STEP	Machining	✓	US$23,000	✓	✓	✓		✓	✓	Website	GPL v3	✓	✓	✓
Alice Exo [87]		STL	3D printing		US$1,600			✓				Website, Hackaday	CC BY-NC 3.0	✓		✓
Mobile Robots																
E-puck [88]	✓	STEP, IGES	Injection molded	✓	US$550–850			✓				Website	Open Source Hardware License v1.0	✓	✓	
Mona [89]	N/A	Unknown	3D printing	✓	$100							GitHub	Unknown	✓		

Miniskybot [90]	✓	OpenSCAD	3D printing		Unknown		✓	✓	✓			GitHub, website, Thingiverse	CC BY-SA 3.0	✓		
Zumy [91]	✓	SolidWorks	Laser cutting		US$400	N/A	✓	✓	✓		✓	Website	Unknown	✓		
Veter [92]	✓	Blender	3D printing	✓	US$1,500		✓	✓	✓			Website, GitHub	GPL v2	✓	✓	
Nanosaur [93]		Unknown	3D printing		US$300	✓	✓	✓	✓		✓	Website, GitHub	CC BY-NC-SA 4.0	✓	✓	✓
ExoMy (ESA) [94]	✓	SolidWorks, STEP	3D printing		US$600	✓	✓	✓	✓		✓	GitHub	GPL v3	✓	✓	✓
NASA's JPL OS Rover [95]	✓	SolidWorks, STEP	3D printing, laser cutting		US$2,500	✓	✓	✓	✓		✓	GitHub	Apache 2.0	✓	✓	✓
TurtleBot3 [96]	✓	Onshape	3D printing	✓	US$549		✓	✓	✓	✓		Website	Open Source Hardware License v1.0	✓	✓	
MIT Racecar [97]	✓	STEP	Laser cutting		US$1,000		✓	✓	✓			GitHub	MIT License	✓		
MuSHR [98]	✓	OpenSCAD	3D printing		US$610–930		✓	✓	✓	✓	✓	GitHub	BSD 3-Clause	✓	✓	✓
Component Modules and Toolkits																
PneuFlex [99]		Unknown	3D printing, molding		Unknown	✓	✓	✓	✓	✓	✓	Website	CC BY-SA 4.0	N/A		
Soft Robotics Toolkit [102]	✓	SolidWorks, STEP	Depends on the project		Depends on the project	✓	✓	✓	✓		✓	Website	Unknown	✓	✓	
NMMI [103]	✓	STEP	Depends on the project		Depends on the project							Website, Source-Forge, GitHub	Depends on the project	✓		
Jasmine [106]		Unknown	Machining		US$130		✓	✓	✓			Website	GPL	✓		
SRoCS [108]	✓	FreeCAD	3D printing		US$7,500 per robot/block		✓	✓		✓		OSF	MIT License	✓		
Digit [112]	✓	STEP	3D printing, laser cutting, molding	✓	US$300	✓	✓	✓	✓		✓	GitHub	CC BY-NC 4.0	✓	✓	✓
Punyo [113]	✓	SolidWorks, STEP	3D printing, laser cutting		Unknown	✓	✓	✓	✓	✓	✓	Website	CC BY-NC 4.0	N/A		✓

(Continued)

TABLE 2. ORH-surveyed projects. (*Continued*)

PROJECT	EDITABLE CAD	CAD SW	FABRICATION METHODS	CAN IT BE PURCHASED?	COST	FABRICATION INSTRUCTIONS	ASSEMBLY GUIDE	BOM/COTS LINKS	ANNOTATED IMAGES	VIDEOS/ ANIMATIONS	TOOLS REQUIRED	ONLINE REPOSITORY	OS LICENSE	SOFTWARE/ CONTROL	COMMUNITY FORUM	UPDATES SINCE 2020
	DESIGN AND FABRICATION					DOCUMENTATION AND INSTRUCTION						DISSEMINATION AND SUPPORT				
Miscellaneous																
Piccolo [119]	✓	N/A	Laser cutting		US$80	✓	✓	✓	✓			GitHub	CC BY-SA 4.0	✓		
FarmBot [120]	✓	Onshape	3D printing	✓	US$1,700–5,300		✓	✓	✓			Website	CC0 1.0	✓	✓	✓
HapKit [121]	✓	SolidWorks	3D printing, laser cutting		US$50–100	✓	✓	✓	✓	✓	✓	Website	CC BY-SA 3.0	✓	✓	
Wooden Haptics [122]	✓	SolidWorks	Laser cutting	✓	US$3,600	✓	✓	✓		✓	✓	Website, GitHub	CC BY-SA 4.0	✓	✓	
ROBO Puppet [123]		Unknown	3D printing		US$85	✓	✓				✓	Website	Unknown	✓		
SPRK [126]	✓	SolidWorks	Laser cutting		US$250		✓	✓	✓			GitHub	MIT License	✓		
ENDO [127]	✓	SolidWorks	3D printing		US$100			✓				GitHub	GPL v3	✓		✓
Pillforge [128]	N/A	N/A	N/A		Unknown	✓	N/A	✓	N/A	N/A	N/A	GitHub	MIT License	✓		
LTA Gondola [129]	✓	SolidWorks	3D printing		US$90		✓	✓	✓			GitHub, Hackaday	CC BY 4.0	✓		✓

OS: open source; HRI: human–robot interaction; OSF: Open Science Framework; MIT: Massachusetts Institute of Technology; VR: virtual reality; GPL: General Public License; OPSORO: Open Source Social Robot Platform; IGES: Initial Graphics Exchange Specification; DyRET: Dynamic Robot for Embodied Testing; ESA: European Space Agency; JPL: Jet Propulsion Laboratory; NMMI: Natural Machine Motion Initiative; SRoCS: Swarm Robotics Construction System COTS: commercial off-the-shelf; GPL: General Public License; MuSHR: Multiagent System for non-Holonomic Racing; N/A: not applicable; SW: software; BSD: Berkeley Software Distribution; PUT: Poznan University of Technology; WDO: wrist-driven orthosis; OMG: open-source Mudra gloves; STEP: standard to the exchange of product data; PARA: printed articulated robotic arm; CASTOR: Compliant Soft Robotics; NICO: neuro-inspired companion; PTC: Parametric Technology Corporation; STL: Standard Tessellation Language robot embodiment that researchers might seek, and novel robot implementations are required to enable expressive capabilities in HRI settings. That said, the number of nonhumanoid social robots trails in comparison to other robotics domains, likely due to the novelty of the field.

HUMANOIDS

Humanoids are distinguished from the social robots discussed in the previous section because they are often used in applications beyond HRI and embodied social behavior. Humanoid platforms have been used in robotics research for studying grasping and manipulation, bipedal locomotion, multirobot collaboration, motion planning, perception, and more. However, building a new humanoid robot is a resource-intensive process and might not be desirable if the research field requires employing the robot simply as a platform upon which to build and validate algorithms. The RoboCup soccer competition spawned several such open humanoid platforms, such as DARwin-OP (subsequently followed by ROBOTIS-OP2 and -OP3) [59] and NimbRo-OP and -OP2X [60], [61] for researchers to deploy in their own specific applications. The hardware and software for all these platforms was released open source to aid users in modifying and repairing the robots, and have since been used in additional research areas of developmental psychology, cognitive science, and education.

Rapid-prototyping fabrication techniques such as 3D printing have enabled users to manufacture components for open hardware more easily, and open humanoid platforms have capitalized on the proliferation of these techniques. The Poppy project [54] [see Figure 3(c)] and InMoov [62] were some of the first 3D-printed open source humanoid robots that could be entirely built and then programmed by the user. Both of these projects have also created a strong community centered around the robot, where users utilizing the robots in different applications can ask questions and share their adaptations and suggestions [63], [64]. The more recent Epi humanoid platform [65] is also assembled with 3D-printed components and focuses on developmental robotics research, with its child-like impression and a fixed base. Similar to Epi, the NICO robot [66] provides yet another platform for multimodal HRI research in developmental robotics and embodied cognition. Finally, the

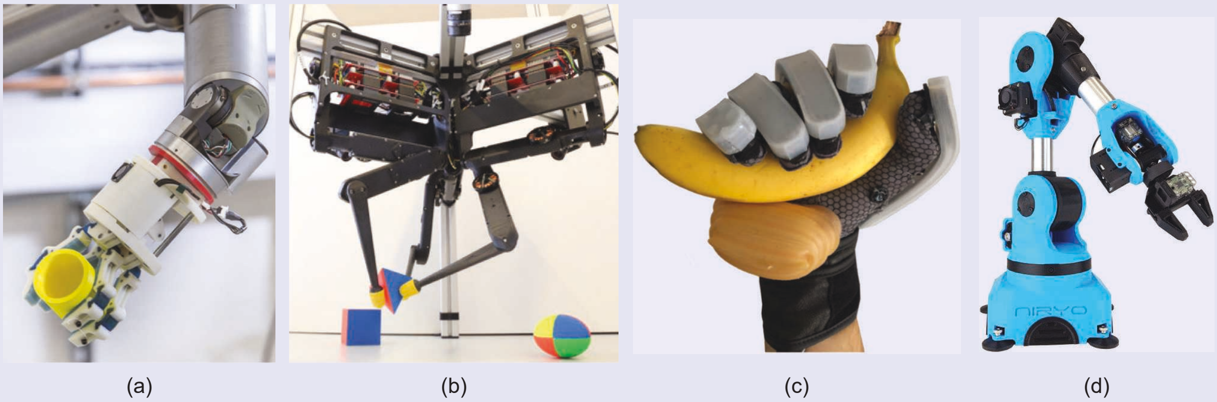

FIGURE 2. Robot arm and hand ORH projects. (a) The model T robot hand from Yale OpenHand [23]. (b) The TriFinger robot platform [24]. (c) The hybrid hand exoskeleton glove [25]. (d) The Niryo One robot arm [26].

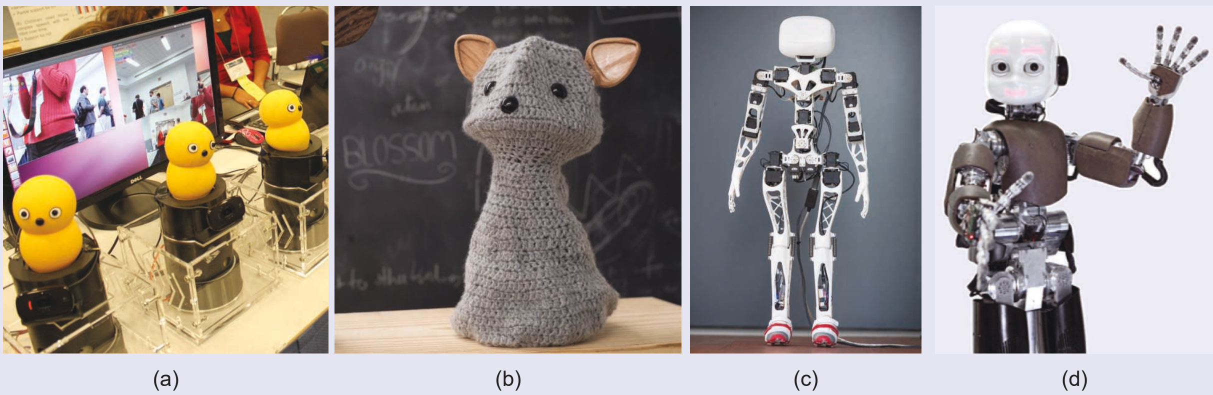

FIGURE 3. Social robot and humanoid ORH projects. (a) The MyKeepon toy robot [52]. (b) The Blossom handcrafted robot [53]. (c) The Poppy humanoid [54]. (d) The iCub humanoid robot [55].

adult-sized humanoid face Eva [67] is designed to communicate using facial expressions and head movements and comprised of 3D-printed skeletal structure and a molded exterior.

One of the most popular humanoid platforms that has found application in several research domains such as locomotion, cognition, haptics, grasping, vision, and learning is the iCub humanoid robot [55] [see Figure 3(d)]. The hardware and software for the robot were open sourced, along with relevant accompanying documentation for both, and has fostered a strong community of iCub users that help resolve issues, share their work, and suggest improvements [68]. The iCub robot is an example of an ORH that is not directly intended for the user to fabricate and assemble in its entirety on his or her own, but the access to documentation and hardware files in the form of CAD drawings and circuit diagrams allows users to modify, service, or upgrade the robot as necessary. Even though iCub was one of the earliest open humanoid platforms and released more than a decade ago, the open source nature of the project has allowed researchers to continue using it as a development platform in an evolving field like robotics, where hardware can fall out of date quickly. The popularity of iCub may have also paved the way for the other open humanoid robot platforms mentioned previously and typifies the benefits of ORH for both users and developers.

LEGGED ROBOTS

Evaluating complex control and learning strategies for legged locomotion in robotics requires testing the algorithms on physical legged robot hardware. But, developing an advanced legged robot can require significant time and resources and custom-fabricated components that might not be easily and inexpensively accessible. Several open source quadruped projects have recognized this challenge and tried to address it by creating robots that use minimal custom-machined pieces, inexpensive components, and off-the-shelf parts. The Open Dynamic Robot Initiative [69] [see Figure 4(a)] was built using low-cost, 3D-printed components and widely available brushless dc motors for the actuator modules. This actuator assembly allows for high-performance torque and impedance control, previously limited to only complex legged robots like Anymal [73] or the Massachusetts Institute of Technology (MIT) Cheetah [74]. Moreover, this same actuation module was implemented in the aforementioned TriFinger robot hand [24]. The other morphologies of quadrupeds that offer a low-cost entry into legged locomotion research include the Stanford Doggo [75] and Pupper and Woofer [76] robots. The Doggo boasts performance metrics comparable to state-of-the-art quadrupeds, while keeping the total cost of the robot below US$3,000. Using a similar architecture, the Woofer and Pupper robots incorporate high power and hobby actuators, respectively, to offer even more options of robust, low-cost legged robot hardware to researchers in this field.

The popularity of Boston Dynamics' quadrupeds has also spawned smaller-scale open source versions such as SpotMicro [77], which in turn inspired OpenQuadruped [78], an inexpensive legged robot with an accompaniment in the PyBullet simulator and OpenAI gym environment to test learned gait policies [79]. Another quadruped platform for legged locomotion study, Oncilla [70] [see Figure 4(b)], enables simplified control in rough-terrain locomotion via compliant, spring-loaded pantograph legs. In addition to releasing blueprints of the robot, a simulated model of Oncilla was also created in Webots for users to test control strategies before executing them on the hardware.

Some quadruped platforms have been developed with a specific application in mind. The Metabot robot [80] was designed as an educational robot platform, enabling educators and learners to experience building a legged robot, and subsequently program it to walk or dance. The Aracna robot [81] was developed to promote research in evolutionary robotic algorithms that can generate robot behaviours automatically. To this end, Aracna deliberately embodies unconventional kinematics that require nonintuitive motor commands for generating a successful gait. Similarly, the Dynamic Robot for Embodied Testing [82] robot was designed to test the evolution of both control as well as morphology of the robot in the real world, and not just in simulation. This four-legged robot is able to modify and reconfigure its legs to adapt to different tasks and environments.

Smaller meter- or centimeter-scale legged robots might be even more suitable than their larger counterparts for an open source release as they are typically fabricated with simple materials and processes and consequently cost less to build. The OpenRoACH [83] hexapedal platform can be built using

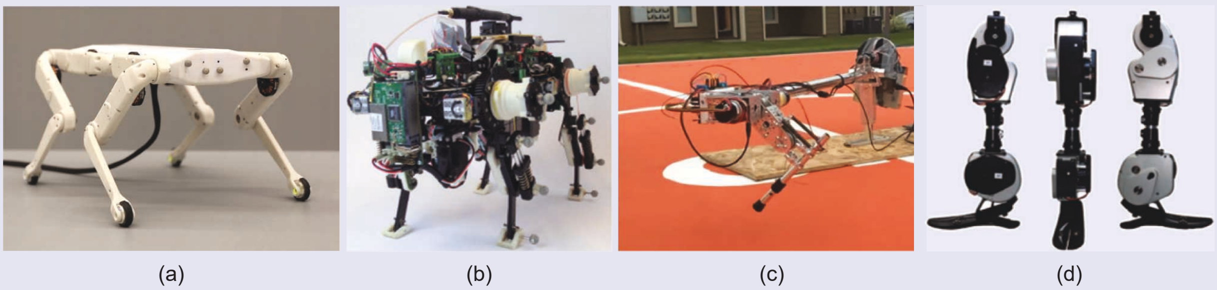

FIGURE 4. Legged-robot ORH projects. (a) The Open Dynamic Robot Initiative quadruped [69]. (b) The Oncilla quadruped [70]. (c) The HOPPY kit [71]. (d) The OSL prosthesis [72].

benchtop machines and ubiquitous rapid-prototyping methods within 2 h. It can be integrated with a variety of sensors, the data from which can be processed through the Robotic Operating System-capable computer on board. Taking the idea of simple fabrication further, the US$20 chassis of the BigANT robot [84] requires only a foam board and minimal hand tools to craft the structure of the robot. The plate and reinforced flexure (PARF) joints in this design can be employed in any number of functional robot designs and facilitates a quick and inexpensive fabrication of robot bodies.

In addition to the aforementioned applications and morphologies, legged robotics research also spans domains of kinematics and dynamic control as well as lower-limb prostheses and exoskeleton designs. The HOPPY kit [71] [see Figure 4(c)] for robotics education allows students to experiment hands on with topics in robotics that range from control and trajectory generation to simulation and kinematics on modular and robust physical hardware. The low cost of the kit makes access easier for researchers and educators studying dynamic behavior in robotic systems. The Open Source Leg (OSL) [72] [see Figure 4(d)] project likewise aims to lower barriers of entry in leg prosthesis research by eliminating the time and resources required in developing one's own robotic leg system from scratch, and presenting a common testbed across the field for benchmarking control approaches. The OSL project also demonstrates some developer practices that can encourage adoption within the community, such as creating an online forum [85] for users to share modifications and ask questions and a step-by-step video assembly guide [86]. Users also have the option to purchase a fully built OSL from a vendor, and the released design and documentation files can help users with future upgrades and repairing the leg. Similar to the OSL, the Alice pediatric exoskeleton [87] was developed with the goal of improving access to lower-limb assistive technologies. Alice can be controlled using simple, low-cost electronics, and the physical components of the exoskeleton are 3D printed or purchased off the shelf, and Alice has even been clinically validated on patients with different medical conditions. That said, lower-limb exoskeletons for adults are still missing from the open source landscape, likely because they would require fabrication with the metal and other high-strength material to sustain the heavier loads during use.

Legged robotics research and education can be cost prohibitive, especially if the robots require advanced manufacturing methods or rely on domain-specific knowledge to build and operate. However, these ORH projects are improving access to physical robot platforms in their respective fields through their comparatively lower cost of entry and minimal fabrication requirements. Across the different robotic leg systems, one can now find an ORH morphology suitable for his or her application, either as it is from the developer or, after some modifications, permitted by the open source license.

MOBILE ROBOTS

Mobile robots are an effective, low-cost, and versatile platform in robotics education and research, and a number of mobile robot platforms at every scale have been proposed, some of which have also been open sourced. One of the most popular mobile robots, E-puck [88] [see Figure 5(a)] was designed with robotics education in mind, and teaches students concepts in signal processing, embedded programming, autonomous control, and distributed systems. The simple structure of the robot, equipped with a microcontroller and a wide variety of sensors, makes for an easy-to-use robot that also has a lot of flexibility for implementation. Although the E-puck robot may be directly purchased from vendors, its part files robot are also publicly available for users to make design changes or build on their own. Other small, two-wheeled open source mobile robots such as Mona [89] and Miniskybot [90] have also been successfully implemented in educational settings, both of which offer lots of flexibility in programming and an inexpensive construction. A tracked mobile robot with 3D-printed structures and mostly off-the-shelf components have also been proposed as benchmarking tools in computer vision and artificial intelligence research in addition to education; Zumy [91], Veter [92], and Nanosaur [93] robots are furnished with more capable onboard computing for users in these domains to test their software on a physical platform, which can be quickly and easily built. To provide students interested in mechanics, electronics, and programming hands-on experience in building and operating a robot, the European Space Agency (ESA) and NASA's Jet Propulsion Laboratory (JPL) have also released their own scaled-down versions of six-wheel Mars rover designs [see Figure 5(b)], along with detailed instruction guides of the fabrication and assembly processes on their websites [94], [95].

Beyond educational applications, robot vehicles have also been used as popular testbeds in a range of robotics research fields. A popular family of robot vehicles, the TurtleBot3 [96], offers a low-cost mobile robot kit that can be adapted to be employed in a range of research applications [see Figure 5(c)]. Although users commonly purchase the TurtleBot3 hardware directly from distributors, developers have released the design files for the robot so that users can customize and modify the robot as needed. A lot of open source mobile robots rely heavily on off-the-shelf components so that the hardware can be effortlessly put together by the user. The Zumy [91] mobile platform is built mostly from off-the-shelf components and is intended for development of multirobot systems and computer vision. Finally, scaled-down miniracecars like the MIT RACECAR [97] and the Multi-agent System for non-Holonomic Racing (MuSHR) [98] can serve as powerful platforms for implementing autonomous navigation, localization, and planning algorithms in real-world environments for robotics research and education [see Figure 5(d)]. But, users can also modify these open hardware cars to adapt the testbeds, for instance, by adding a gripper to the MuSHR to investigate collaborative multirobot manipulation [98].

Mobile robots serve as a very effective tool for testing implementations of software algorithms in a range of robotics domains quickly with a reduced cost of materials and a simple construction requiring nominal expertise and tools. Their ease of use and adaptability to different applications also make them suitable for educational settings in addition to robotics research.

COMPONENT MODULES AND TOOLKITS

In some of the newest domains of robotics, building one's own hardware might be the only option available to a researcher because no similar off-the-shelf products are commercially available for purchase. For such domains, ORH can provide valuable tools for users to iterate and test new robot structures instead of developing hardware from scratch.

Soft robotics is one of these fields in robotics that has grown rapidly over the last few years, and researchers have been open sourcing their hardware to allow other researchers to create their own soft robotic devices. One such core component in a soft robotic mechanism is the actuator, and PneuFlex [99] was one of the earliest soft actuators that developers released publicly, along with CAD models, 3D printing files, and a detailed fabrication guide [100]. The actuator was later used in the compliant Robotics and Biology Lab hands [101]. A lot of soft robot component technologies have also been compiled on open platforms that host tools and guides to test new soft mechanisms quickly, further accelerating the development of novel applications for soft robots. The Soft Robotics Toolkit [102] offers instructions and guides on several common components of soft robots to enable speedier development of soft devices, as shown in Figure 6(a). The toolkit was developed with contributions from a number of research groups and has even been used to support educators in hands-on robotics courses. Another such platform supporting the fast development of articulated soft robots is the Natural Machine Motion Initiative (NMMI) [103]. The NMMI is composed of hardware modules that users can put together to create complex and novel robotic structures. Both of these platforms are continually updated with the latest developments in soft robotics and grow with more contributions from developers and other research groups.

The practice of open sourcing the tools and components required to build a robot hardware has gained traction not only in soft robotics but has also seen prevalence in other evolving domains of robotics, such as modular and swarm robotics. The Molecubes [104] platform was openly distributed to promote new developments in reconfigurable and modular robotics. The project underwent multiple iterations and invited modifications and enhancements from the user community [105]. Swarm robot hardware has similarly been made publicly available in projects like the centimeter- or millimeter-sized Jasmine [106] and Kilobot [107] robots. The Swarm Robotics Construction System (SRoCS) [108] is another open source swarm robot platform that was released for studying multirobot coordination, specifically in construction tasks [see Figure 6(c)].

Finally, hardware that spans multiple areas in robotics can also significantly benefit from open source technologies. Electronics platforms like Arduino [109] and Raspberry Pi [110], common components such as tactile-sensing arrays like TakkTile [111] and Digit [112], and soft tactile grippers in Punyo [113] are constituent in robot hardware for many different areas [see Figure 6(d)]. Electronics hardware, in particular, has a long, sustained history of open source communities and repositories like Open Electronics [114], Kitspace [115], Hackaday [116] and many more and have been surveyed in field-specific academic reviews [117] and special issues of journals [118]. Although the wide breadth of open source electronic platforms and components is beyond the scope of this review, their widespread adoption and thriving community can inspire best practices and methods in disseminating ORH projects. Moreover, the success of these repositories also confirms that ORH projects that release component modules and technologies can notably expedite prototyping and testing of new robot structures, which can be especially vital for identifying novel applications in fast-moving domains of robotics.

MISCELLANEOUS

CNC ROBOTS

Open source hardware has been pervasive in many other robotics domains in addition to the ones outlined previously. CNC robots are so called for their similarity to CNC machines. These simple three-axis robots with orthogonal actuation DoF are potent tools for automating a variety of 2D tasks. They can be equipped with any suitable end effector for the task. For instance, the tiny Piccolo CNC bot [119] applies this architecture as a drawing robot with a distal pen attachment [see Figure 7(a)]. Piccolo is often used in educational workshops to give users hands-on experience in building and programming

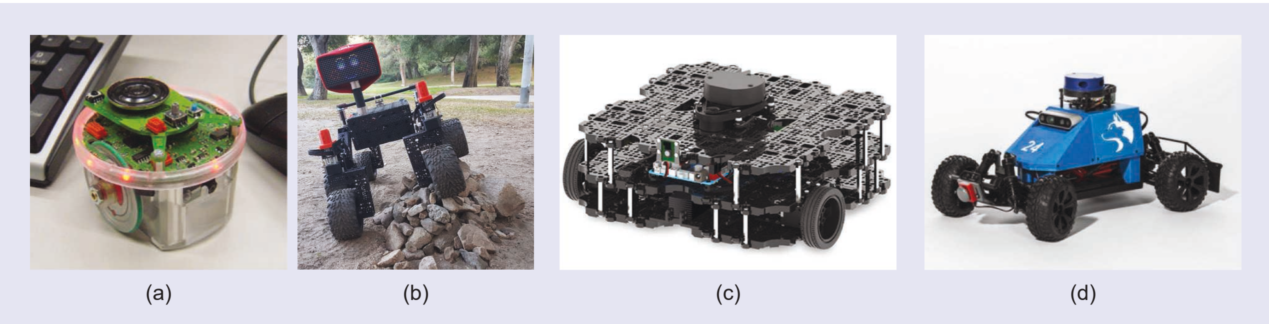

FIGURE 5. Mobile robot ORH projects. (a) The e-puck two-wheeled robot [88]. (b) NASA's JPL Open Source Rover [95]. (c) The TurtleBot3 Waffle platform [96]. (d) The MuSHR miniracecar [98].

robots. The FarmBot [120] employs the CNC robot framework to automate small farms in home agriculture spaces [see Figure 7(b)]. The developers of FarmBot sell the hardware, but the source files for the robot are freely available and can be modified or repaired as per the users' needs.

HAPTIC AND TELEOPERATION INTERFACES

Haptic and robot teleoperation interfaces have also seen ORH solutions that users can build on their own. The HapKit [121] platform is a 1-DoF device that can be used as an input and also generate forces that are experienced by the user [see Figure 7(c)]. The kit is quite inexpensive (costing less than US$50) and can be intuitively built and used by a novice user, which is ideal for learning environments. Having a similar goal of being easily fabricated, WoodenHaptics [122] is a 3-DoF haptic device, the structure of which is made from laser-cut stacked plywood sheets. The design of the device also focuses on the ease of making modifications to the device for different applications requiring spatial haptic interactions. Although haptic devices have been commonly used as interfaces in teleoperating a robot, the ROBOPuppet [123] takes an alternate kinesthetic approach to simplifying teleoperation by creating a scaled-down replica of the target robot. The operator is expected to be able to more intuitively control the motion of the robot by manipulating the tabletop robot model, whose joint angles are duplicated exactly in the target robot.

MEDICAL ROBOTS

Open sourcing hardware in medical robotics can be challenging, particularly because of the stringent requirements placed on robot systems in this field. Although open source platforms such as the da Vinci research kit (DVRK) [124] and Raven-II [125] have markedly improved accessibility to the control software of commercial surgical robots, the hardware of medical robot systems remains relatively more exclusive. Some ORH in medical robotics targets peripherals around the open software surgical robots, such as the Stewart Platform Research Kit platform [126], which simulates body organ motion for studies with the DVRK robot. Medical robot hardware itself has started to become publicly available through initiatives like Open Source Medical Robots [127] and Pillforge medical capsule robots [128]. The ENDO robot presented in [127] is a continuum manipulator designed to lower the barrier of entry in surgical robotics research. And in [128], a platform for rapid development for medical capsule robots is proposed, with hardware and software component modules that can be assembled for testing prototypes quickly. However, ENDO and Pillforge are just a few ORH solutions in medical robotics, and more open source hardware that can further lower barriers to new robotic technologies in this field should be explored.

AERIAL ROBOTS

ORH for UAVs has been widely popular among researchers for their flexibility and ease of implementation, similar to ORH for mobile robots, as discussed previously. Lim et al. [12] present a survey of open hardware for quadrotor UAVs, so for brevity, those projects are not detailed here. Hardware for other types of UAVs, like lighter-than-air (LTA) crafts, have also been open sourced; a design for a robotic gondola attached beneath a balloon is proposed in [129] as an indoor LTA airship that can be employed in research and education [see Figure 7(d)]. Similar to medical robots, open source projects in UAVs have been explored more as software platforms and flight controllers. But as more applications for UAVs are discovered and, with the increased convenience of fabrication from rapid-prototyping methods, ORH for UAVs can be expected to also become just as abundantly available as mobile robots.

DEVELOPING AN ORH PROJECT

Many of the ORH projects categorized in the previous section exemplify a development process that can lead to a widely used and retained ORH with an active user community. Prospective developers can better understand the steps involved in creating an ORH through these projects, especially if they work in a similar robotics domain. In the following discussion, we detail the pipeline of developing a successful

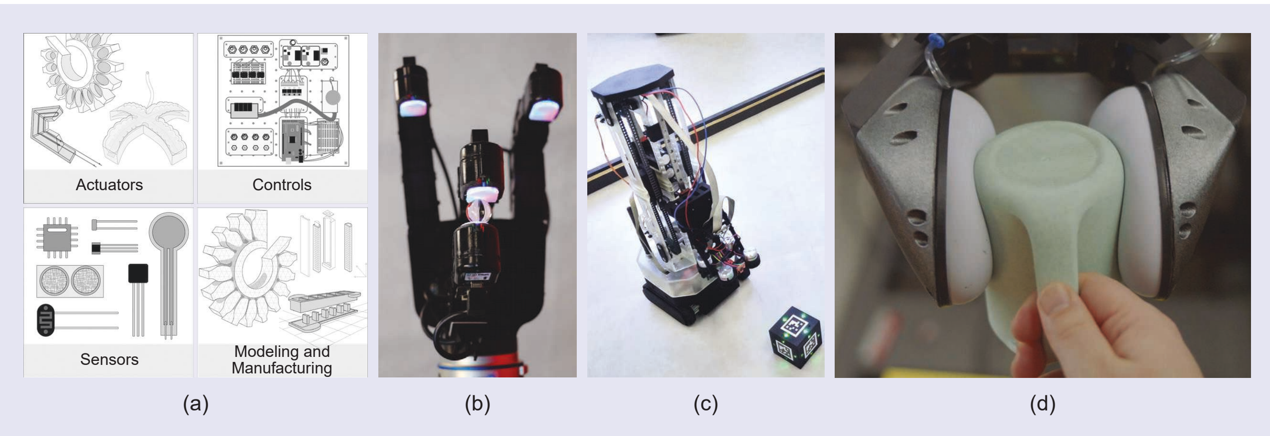

FIGURE 6. A component module and toolkit ORH projects. (a) Components of the Soft Robotics Toolkit [102]. (b) A digit sensor on a multifingered hand [112]. (c) The SRoCS [108]. (d) A Punyo soft bubble gripper [113].

ORH through the stages of design, documentation, and dissemination, which will help an ORH project become widely adopted by many users for a long period of time. We also discuss how developers can determine fabrication methods most suitable for their ORH. And along the way, we refer to some useful strategies from ORH projects previously surveyed that new developers can implement themselves.

DESIGN AND FABRICATION

DESIGN AND PART FILES

An ORH project vitally shares the design files for the hardware so that users can manufacture components as well as modify and share their adaptations. To this end, the developer should publish all part documents, including 3D CAD files (for any machined, 3D-printed, or molded parts), 2D fabrication files (for laser cutting, waterjet cutting, or similar process), and any auxiliary files required for fabricating components (for example, machine drawings or circuit board layouts). If any of the part files have special software requirements, the developer should highlight those in the shared documentation. As such, open source projects should rely on freely available software so that the cost of meeting the software requirements do not prohibit users from accessing the released design and part files. That said, the majority of ORH are developed in proprietary CAD systems like SolidWorks or Autodesk Fusion, usually due to the lack of free, advanced 3D modeling software, which are suitable for the design complexity of some mechanical components. Nonetheless, projects like Metabot [80], MuSHR [98], Veter [92], and Miniskybot [90] use free and open source 3D modeling tools like OpenSCAD, FreeCAD, or Blender, and others like FarmBot [120] use free versions of design platforms like Onshape. Even if editable design files are made with these freely available modeling tools, they might still remain out of reach of many users who do not have access to expensive computers capable of opening and modifying large projects with lots of components and subassemblies. In contrast, programs like Onshape and Autodesk Fusion have the benefit of being cloud based, which eliminates the requirement of having sufficient local computing power to run CAD software, and allows users to flexibly work on design files from any computer with a good Internet connection.

It is worth noting that a simple replication of an ORH may not require editable CAD files. For instance, if a robot requires only 3D-printed components, STL files may be sufficient. However, editable 3D CAD models of the components allow users to adapt designs and make changes as necessary, which can then be shared with the rest of the user community or even released as another ORH. Editable CAD files could be in formats directly saved by programs (like .sldprt from SolidWorks). But most of the major CAD programs are not able to easily import file formats native to other programs. So, designs can be made even more accessible by exporting files to neutral-interchange formats like .step or .iges because of their cross-platform compatibility. The .step format is regularly updated to include more solid model data (such as geometry, configurations, colors, layers, geometric dimensioning and tolerancing, and more) and is thus preferred over the older .iges format, which retains surface-model data only. The Easy-Hand [28] project is an example of ORH redistribution after modification of the Yale OpenHand project [23] because the 3D models were made available for OpenHand. The sharing of files in modifiable formats also applies to 2D part files, such as laser or vinyl cutting files as well as electronics components of the hardware, such as schematics for circuit board layouts.

To encourage adaptations of the design and contributions from users, the features in part files should be added such that they clearly convey purpose and function of the design choices. Breaking the top-level intricate assembly down into simple, understandable subassemblies can further help improve the ease of parsing through the files. Both the ESA's ExoMy [94] and NASA's JPL Open Source Rover [95] have an involved, top-level assembly with several off-the-shelf components and 3D-printed parts. The ExoMy project details a part-naming convention for its mechanical design files, along with a folder structure guide to navigate the assemblies and 3D-printed parts. The Open Source Rover provides a similar folder structure guide but also outlines a subsystem road map to provide users with a high-level view of the subassemblies, and at which point along the fabrication and assembly process the user is at any time [95].

(a) (b) (c) (d)

FIGURE 7. Miscellaneous ORH projects. (a) The Piccolo CNC bot [119], (b) FarmBot CNC robot [120], (c) HapKit haptics platform [121], and (c) robotic gondola for a lighter-than-air (LTA) airship [129].

Another reason for ORH projects to share editable design files is to account for inconsistencies in fabrication processes. Processes like 3D printing vary depending on the printers and materials used, and developers should consciously avoid designing hard-to-fabricate features in their custom parts. One way to avoid such variations altogether in fabrication processes is to rely more on commercial off-the-shelf (COTS) components. But, even for purchased components, developers should consider alternatives to components that may not be in stock or discontinued in the future. Users with access to editable CAD files can tweak the part designs on their end to accommodate alternative components in such cases. This allows the developer to offload creating design variations to the users, who can add compatibility for different COTS parts on their own, given that the design files are understandable and the file repository is easily navigable.

FABRICATION METHODS SELECTION

One of the main hurdles that open source hardware has to overcome is that it requires fabrication of physical components to reproduce the hardware. So, the fabrication methods that developers choose for their components can have a significant impact on the accessibility of their ORH project. Developers should identify the requisite processes in parallel with the design of the components and modify any parts that might be hard to manufacture or source. Many ORH projects heavily rely on additive manufacturing approaches such as fused deposition modeling and more modern techniques like stereolithography, polyjet, and digital light processing. These are quite approachable today, even with desktop 3D printers, and are generally appropriate for small parts that may have complex geometries. The range of materials that can be 3D printed has grown considerably in recent years [130], but most of these printed polymers are not suitable for very high-force or long-duration applications. Metal 3D printing is starting to become more accessible, particularly through on-demand prototyping services, but it is still limited in terms of material choices and is sometimes cost prohibitive [131]. There are also several other fabrication techniques that use 3D printing as one of its steps. Printed parts can be used as molds for different resins and are commonly used to make components for soft-elastic actuators [132]. Other techniques have taken the idea of molding with 3D-printed parts even further, such as shape deposition manufacturing [133], which alternates deposition and removal of material to create embedded structures with different materials. HDM [22] builds on this concept by including both permanent and sacrificial parts in the mold and has been used in making the robot's grippers and hands of the OpenHand [23] and OpenBionics [29] projects.

For larger parts, developers can consider subtractive methods such as laser cutting, laser etching, CNC routing, and waterjet cutting. These are often very fast, inexpensive, and can work with a variety of materials. Although they are limited to extruded planar geometries, many projects have found ways to fabricate components using these methods. Piccolo [119] and an early HapKit version [121] use mostly laser-cut acrylic or fiberboard, the legs and frame of the quadruped Stanford Doggo [75] are made with waterjet aluminum sheets, and the Pupper [76]. The skeleton of the OPSORO robot [56] is also mostly composed of laser-cut foam pieces that can be snapped together. These pieces are also designed with assembly error-proofing in mind, for example, with connectors of different widths that fit together only in a specific way. Another laser-cutting-based technique stacks multiple layers of cut 2D parts to create thick structures, as seen in WoodenHaptics [122]. The OpenRoACH robot [83] is made from etched wooden sheets with flexure joints that serve as creasing patterns so that the 2D sheets can be folded to create 3D robot structures. Laser-cut parts have also been used as stencils for painting different dot patterns onto latex sheets in the Punyo tactile gripper [113]. Even without laser cutters or waterjet cutters, rigid sheets of foam, plastic, cardboard, fiberboard, or other material can be scored and cut with common tools to easily construct structures for robots. Fabrication of the BigANT robot [84] presents a versatile technique (called a *PARF* fabrication) of building robots with rigid plate materials connected by fiber-reinforced tape joints. These types of versatile fabrication methods that rely on minimal tooling and less expensive materials can help ORH projects reach a much wider audience of users.

Depending on the functional requirements of the components in a robot, certain material properties may be desired, which then inform the choice of fabrication method. Soft robots use a number of different approaches to manufacture elements of their robotic systems [134]. Many social robots are also designed with soft, deformable materials because they need to be compliant to external contacts, or blend with household objects. The Blossom [53] robot's exterior is made from knitted fabrics that are used to convey warmth as the robot is intended to be stationed in people's homes. The OPSORO robot [56] also uses flexible textiles as well as foam patterns that are stitched together to make the 3D shell. Just as some robots need highly deformable components, others may need high-strength components to sustain large loads. The OSL [72] is one such project that requires several machined metal parts. But machine tools like CNC mills and lathes might be out of reach for many users who may not be trained to use them. To mitigate that, the OSL project offers the option of purchasing a prebuilt leg or suggests that users outsource the fabrication to machine shops. When possible, developers should still try to recommend easy-to-use fabrication methods for their components. And if no existing method is suitable, they could rely on COTS components such as extruded frames, tubes, or patterned plates, which can be readily adapted into large-size robot builds. For example, FarmBot [120] and HOPPY [71] both use metal extrusions; linkages of the PARA robot arm [50] are made from acrylic tubes, ROBEL [31] is mostly composed of off-the-shelf brackets, and Zumy [91] and HOPPY are mostly made from COTS parts.

Even for users with no access to or expertise with manufacturing equipment, obtaining custom-fabricated parts is becoming easier with the rise of on-demand prototyping and manufacturing services [135]. Developers could utilize the wide range of manufacturing options available with popular services (like Protolabs, Shapeways, Fictiv, Xometry, and many more) to recommend to users as an alternative to fabricating parts on their own.

DOCUMENTATION AND INSTRUCTION

The design and part files on their own do not suffice for easy reproduction of the ORH. Thorough documentation, along with visual guides for fabrication and assembly, are essential to ensure accessibility for users, with a range of expertise in the processes required to build and operate the robot hardware. The ORH projects that have been widely adopted have clear and detailed instruction sets that indicate each step of the manufacturing and assembly process. The documentation package of the OpenHand project illustrates some of these good documenting techniques. A list of all the fabricated and purchased components in the form of a BOM is listed at the top of each section of the assembly guides, as shown in Figure 8 [27]. The instructions for building OpenBionics hands [34] is also exemplary of some best practices [136]. For instance, each section has an accompanying list of materials and annotated images of the parts for users to quickly reference assembly steps and the requisite components at hand. The BOM for the Piccolo bot [119] and Punyo gripper [113] help link where each component can be sourced, or alternately, specify the fabrication process if it is a custom component [137], [138]. However, a user might have acquired all the requisite components and realize midway through the build process that he or she does not have a particular tool. Projects like Poppy [54], WoodenHaptics [122], and ExoMy [94] avoid such a situation by listing and visually showing all the tools and equipment required to carry out the assembly [139], [140].

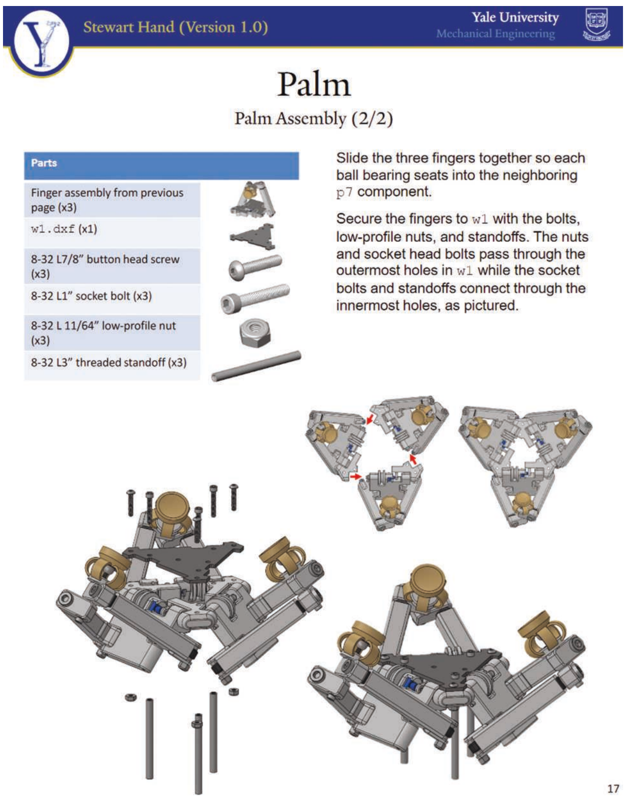

FIGURE 8. The BOM visually shown and annotated images of the exploded assembly view in the documentation of Yale OpenHand's Stewart hand [27].

Once all the components and tools are obtained by the user, the documentation should then direct he or she through each step of the assembly process. Visual guides are very effective in conveying the process, in addition to textual instructions. The InMoov project uses images of the partially assembled components after each step for users to compare their output with that in the guide [62]. A number of other projects, including the Yale OpenHand [23], Punyo gripper [113], OSL [72], Poppy [54], and more, have videos of an individual assembling the components so that users can conveniently follow along, e.g., [141] shows videos for fabricating an actuator body in the Soft Robotics Toolkit [102]. Such assembly videos, even more so than still photos, can unambiguously communicate the minute details of the process, compared to just reading the text; for instance, the user may pick up on how much clamping force to apply through the individual's actions in a video. The HOPPY project [71] and CASTOR robots build instructions [58] instead of using video animations of the various CAD model's exploded views, which also effectively articulate how parts connect with one another.

Inconsistencies at the user's end depending on the type of fabrication equipment, tools used for assembly, or alternative off-the-shelf components sourced are likely to occur and need to be addressed when the developer prepares the documentation packet for the ORH. The previous section detailed how the design of the hardware can account for such variations. The documentation and instruction guides can also aid the user's ability to identify discrepancies by highlighting which design features the user should check before proceeding with the assembly, and then to compare with the developer's setup for differences (for example, make of the 3D printer or model number of the actuator). The Yale OpenHand instruction documents have annotated images and schematics to call the user's attention to evaluate specific features on his or her molded and 3D-printed parts [27]. This helps debug any issues in fabrication early on in the assembly so that users can reproduce well-functioning robot hands.

Developers are also strongly encouraged to provide as much information to users as possible about the appropriateness of various fabrication technologies, including specific machine models used for the project's components. For example, the ExoMy rover project lists all the 3D printers that its rover has been successfully manufactured with [142], and users are encouraged to notify the developers if they printed the rover with a different 3D printer so that the developers can update this list. For more multistep fabrication processes, such as those for the Punyo gripper [113], the documentation thoroughly guides the user through the process steps and, most importantly, includes safety precautions and fabrication errors to look out for along the way [143]. The projects that require more advanced fabrication steps, like CNC-machined parts, should be aware of the level of complexity and cost that would discourage potential users. The OSL project [72] has several machined parts due to functional requirements of the hardware, but the developers provide all the necessary files to outsource the machining fabrication, such as drawings, part quotes, and even sample e-mails to send to manufacturers.

Thoughtfully created documentation packages are thus critical to accompany the chosen fabrication technologies. The practices typified in the aforementioned ORH projects significantly lower the barrier of entry for novice users, and they should be referenced by prospective developers intending to create ORH documentation that improves the reproducibility of their hardware.

DISSEMINATION

The key characteristic of an ORH project is that the prepared design files and documentation packet are disseminated in an open manner for potential users to find and access them. For this purpose, many ORH projects host project repositories on platforms like GitHub or the Open Science Framework, which allows public dissemination of the project, along with an ability to version-control files for any necessary updates. More hardware-focused platforms like Thingiverse and Hackaday are popular with maker communities but have also been used for distributing new research hardware [29]. Wherever the developer decides to host the project files, they should track revisions and updates to convey changes in the latest version to users. In fact, the developer should try to ensure proper maintenance of the ORH project over its active duration of time by releasing patches and fixing persistent issues that users may be encountering. To publicize an ORH, developers can utilize academic journals on open source hardware such as HardwareX [18], *The Journal of Open Engineering* [19], and *The Journal of Open Hardware* [20], which can serve as valuable tools for broadcasting one's work within the research community. ORH projects such as the Eva humanoid face [67], HRI hand [32], and SRoCS [108] were all published through these open hardware journals. Other dissemination strategies for advertising ORH specifically to reach academic and research communities can also be used, such as sending e-mails to popular robotics mailing lists, publishing papers in field-specific journals, and participating in conferences, workshops, and tutorials. Developers should also contact directories like the ORH initiative [16], which compile ORH projects. OSHWA [17] maintains a similar directory of open source hardware projects but requires developers to undergo a certification process beforehand. Certifying the project with OSHWA ensures that it complies with the requirements set for their open source hardware standard, further adding credibility to an ORH with the OSHWA certification logo.

"EXPLICIT MENTION OF THE LICENSE ON THE PROJECT'S REPOSITORY AND DOCUMENTATION CONVEYS TO PROSPECTIVE USERS HOW THEY CAN IMPLEMENT AND USE THE WORK AND SHOULD THUS BE CAREFULLY CONSIDERED BY THE DEVELOPER."

When releasing an open source project, developers need to decide which type of license should be applied to their original design files as well as the accompanying software and documentation. Different parts of the project can have distinct licenses applied to them; for example, the BOM for HRI hand [32] and SRoCS [108] note the open source license next to each part in the table. The applied license stipulates whether users need to attribute the original work (attribution), whether they can modify or adapt the work (derivative), whether they can distribute the adapted work under a different license (permissive, as opposed to copyleft or viral), and whether they can use the original open source work for a monetary advantage (commercialization). There are several open source licenses with different characteristics that developers can choose from to apply to their designs [144]. Some of the most commonly applied licenses are from Creative Commons [145], which has several licenses with different provisions on attribution, use of derivatives, copyleft, and commercial use. A comprehensive comparison of the Creative Commons licenses can be found in [145] and [146]. Popular open source software licenses like GNU General Public License (GPL) [148], MIT license [149], and BSD license [150] are often applied to open source hardware, primarily as a way to waive liability and warranty. They differ from each other in one major way: the GPL is copyleft and requires any future modifications to be released under the same license, whereas the MIT and BSD licenses are far more permissive and do not have many licensee requirements. The Apache license [151] is similar to the MIT one but has explicit provisions for patents on derived projects. Although all of these licenses are written with software projects in mind, some licenses have also been written to specifically protect hardware. Both Tucson Amateur Packet Radio [152] and Centre Européen de Recherches Nucléaires (CERN) [153] have copyleft licenses for hardware similar to the GPL (CERN even has different variants), and the Solderpad license [154] is a permissive hardware license derived from Apache.

Clear and explicit mention of the license on the project's repository and documentation conveys to prospective users how they can implement and use the work and should thus be carefully considered by the developer. Some repositories like GitHub even provide the option to select a license when creating a new project. In general, developers should include the license information in every file or at least in the README [17]. Similar licensing principles also apply to nonhardware files of the project, such as any accompanying code and even supporting documentation packages, and developers can choose to apply different licenses to each of these elements. Overall, it is strongly encouraged that developers pick an open source license with as few restrictions as possible to attract users to their projects [155] and consider that many users would like to customize and build upon the project.

SUPPORT

Previously, one of the challenges identified by users of open source hardware was the lack of sustained support for technical issues that users may encounter during the build process or in later stages of operating the hardware. Although developers should nurture ORH projects through updates that resolve issues encountered by users, some of this responsibility can be undertaken by an active user community. Repositories or project websites should at least allow users to contact developers to report issues and suggest modifications. But a thriving user community can be generated if developers create a forum space for users to log issues, ask questions, share updates, and even offer support to each other. Popular ORH projects like iCub [55] have a strong community forum, with many users of the iCub robot participating in discussions ranging from low-level software questions to announcing new research conducted with the robot [68]. The ExoMy [94] and NASA's JPL rover [95] projects have similarly set up forum spaces and message boards for users to seek solutions to common problems as well as invite contributions to the project by proposing changes and updating project files. Robotics toolkits and component platforms like Soft Robotics Toolkit [102] and NMMI [103] in particular benefit from highlighting opportunities to their communities to contribute to their projects because these improve the collection of modules offered on the platform. Several other ORH projects like WoodenHaptics [122], InMoov [62], and FarmBot [120] allow users to interact with each other through mailing lists, discussion spaces, and blogs, which can lead to a more engaged user base that subsequently helps the project stay relevant to progress in the wider robotics community over a longer duration through "cooperative development" [156]. These practices facilitate the creation of a thriving and engaged user community for a successful ORH, and discussion forums and message boards are cornerstones in building and maintaining cooperation in the project's community. Through these communication channels, users can share feedback, frequently encountered issues, contribute design updates and modifications, share new project ideas and implementations, and offer support to each other, all of which enhance the ORH's usability and compatibility far more effectively in comparison to only the developer working to sustain those ORH attributes.

"ALTHOUGH DEVELOPERS SHOULD NURTURE ORH PROJECTS THROUGH UPDATES THAT RESOLVE ISSUES ENCOUNTERED BY USERS, SOME OF THIS RESPONSIBILITY CAN BE UNDERTAKEN BY AN ACTIVE USER COMMUNITY."

THE FUTURE OF ORH AND CONCLUSIONS

With just the sheer magnitude of projects that are released every year, the momentum behind ORH will certainly be sustained over the coming years and still has the potential to revolutionize access to new hardware in robotics [21]. That said, releasing robotics hardware in an open source manner is still far from standard, and trails behind open source software and electronics. But robotics research can greatly benefit if open sourcing hardware become a more common norm, as is currently the case with published algorithms and software. Computational researchers release their software code for purposes of reproducibility, benchmarking, verification, and more [157], and releasing hardware that can be replicated by others would promote research in the same way. This is evident in emerging fields of robotics, such as social and soft robotics, where increasingly, projects are being released open source. Commercially available hardware is somewhat scarce for these new fields, which may also instigate researchers to release their projects as well as seek out other researchers' ORH. But even beyond these novel areas, the proportion of ORH will continue to grow, facilitated by an intersection of the new domains with the established fields of robotics, for instance, as soft actuators make their way into grippers, mobile manipulators, and assistive devices.

The future of ORH will also be fueled by the shrinking barrier of entry for novice users and the steadily improving quality of available ORH. The practical challenges of ORH, for both users and developers, are already being eased up by technological advancements on several fronts. First, new fabrication techniques are more accessible than ever and are able to generate complex and durable parts in a variety of material options, such as metals and soft polymers. On-demand manufacturing services have also helped outsource the fabrication step for users without access to costly equipment, further boosting the accessibility of ORH in many domains [135]. Second, free and open source CAD software is becoming more feature rich, especially cloud-based CAD, which makes creating and modifying 3D parts significantly easier without the need for expensive computers [158]. Next, communication spaces, like online forums set up on Slack or Discord, are more ubiquitous today and being used more often by developers, which encourages collaborations and future development of the project [156]. And finally, repositories for disseminating project files are being more commonly utilized, and these platforms continue adding functionality for managing and controlling versions of design and documentation files. Even the more hardware-focused platforms such as Thingiverse and Hackaday, which were previously popular in open source communities, are gradually being used for sharing new ORH projects [29], [48], [62], [77].

Although the quality and abundance of ORH are important, future ORH projects also need to be supported for a sufficiently long duration of time. The reliability of long-term access and support is one of the reasons that drives prospective users away from open source hardware. But the growing utility of development platforms for collaboration, hardware repositories, and project management tools can aid in extending the life of future ORH projects [147]. These platforms help invite new users, engage experienced users, and encourage potential developers to share their own adaptations. In this way, sustaining and continually iterating the ORH is distributed within the community and simplified for the developer. Eventually, this cycle of sharing ORH can also stimulate more sharing, thus compounding the advantages for both users and developers.

> "THE FUTURE OF ORH WILL ALSO BE FUELED BY THE SHRINKING BARRIER OF ENTRY FOR NOVICE USERS AND THE STEADILY IMPROVING QUALITY OF AVAILABLE ORH."

One of the crucial areas in which the ORH community needs to invest more resources and effort in the future is in properly documenting and updating their projects. From the surveyed projects listed in Table 2, only roughly a third of the projects have been updated in the last two years. Although the repositories may still be accessible and have ample supporting documentation, dormant ORH projects are unlikely to be adopted by users, especially if there is no active community or forum around the project. Moreover, robotics is a field that moves quite fast, and the longer a project remains dormant, the more likely it is to become obsolete. Some other vital aspects of the documentation and instruction/user guide missing in many of the existing ORH projects are detailed fabrication instructions and a comprehensive, regularly updated BOM. A few do offer the alternative option of purchasing the components from a vendor, but the majority of ORH projects do not have much guidance on how to fabricate or acquire the required components. This can be a critical barrier to adoption of an ORH project, particularly for users who might be new to a field, or for prototyping methods that might not yet be widely used. Thus, in the future, the role ORH projects play in uplifting robotics research will be vastly amplified if they are regularly supported and updated over a much longer time period and accompanied by a comprehensive documentation package that covers a wide range of instructions from fabrication to operation.

CONCLUSIONS

In this review, we focused on ORH, which we defined as projects having open sourced their design and auxiliary files accompanied by proper documentation, addressing a need in some robotics-specific domain, and centering around mechanical or electrical hardware components. The characteristics of ORH projects were discussed by highlighting the advantages and challenges encountered by both the users and developers of ORH. For the user, open source projects allow flexibility to adapt the hardware, and to make modifications, repairs, and upgrades with support from the wider user community. On the other hand, the replication of ORH requires the user to have some knowledge of fabrication and assembly, which can be especially challenging for projects without good documentation. For the developer, open sourcing the project ensures reproducibility of his or her hardware, verification of the designs, and valuable exposure to the community when the hardware is widely adopted. But developing an ORH can be quite demanding too, requiring preparation of supplemental documentation and instruction guides, and maintaining simplicity in the designs. Furthermore, the different domains within robotics can necessitate unique hardware and distinct design and fabrication methods, further diversifying the range of ORH projects.

To both summarize the state of the art as well as investigate best practices from current ORH, we surveyed more than 80 projects and classified them by their application domain within robotics. Although not an exhaustive list, the survey aims to highlight the properties of ORH that are characteristic to each of the domains. Moreover, the strategies applied by popular ORH projects classified are noted. These best practices and common fabrication methods are then relayed so that developers can follow them through the design, documentation, and dissemination phases of new ORH projects. We will make these ORH features, development practices, and the survey available online on the ORH website [16].

Open source projects have proliferated throughout robotics, aiding progress and innovation from accelerated implementations of novel ideas. This has been evidenced by the growing availability and widespread adoption of open hardware over the recent years. We believe that ORH will continue to have a lasting and notable impact across the robotics landscape, and its development warrants further exploration efforts.

ACKNOWLEDGMENT

This work was supported by the U.S. National Science Foundation under grants 1734190 and 1928448.

AUTHORS

Vatsal V. Patel, Department of Mechanical Engineering and Materials Science, Yale University, New Haven, CT 06520 USA. E-mail: v.patel@yale.edu.

Minas V. Liarokapis, New Dexterity research group, Department of Mechanical and Mechatronics Engineering, University of Auckland, Auckland 1010, New Zealand. E-mail: minas.liarokapis@auckland.ac.nz.

Aaron M. Dollar, Department of Mechanical Engineering and Materials Science, Yale University, New Haven, CT 06520 USA. E-mail: aaron.dollar@yale.edu.

REFERENCES

[1] C. Pantofaru, S. Chitta, B. Gerkey, R. Rusu, W. D. Smart, and R. Vaughan, "Special issue on open source software-supported robotics research," *Auton. Robots*, vol. 34, no. 3, pp. 129–131, Apr. 2013, doi: 10.1007/s10514-013-9325-4.

[2] A. Staranowicz and G. L. Mariottini, "A survey and comparison of commercial and open-source robotic simulator software," in *Proc. 4th Int. Conf. Pervasive Technol. Related Assistive Environ.*, New York, NY, USA: Association for Computing Machinery, 2011, pp. 1–8, doi: 10.1145/2141622.2141689.

[3] M. Quigley et al., "ROS: An open-source robot operating system," presented at the ICRA Workshop on Open Source Software, Kobe, Japan, 2009.

[4] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *Int. J. Adv. Robot. Syst.*, vol. 3, no. 1, p. 8, 2006, doi: 10.5772/5761.

[5] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–123, 2000.

[6] H. Bruyninckx, "Open robot control software: The OROCOS project," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2001, vol. 3, pp. 2523–2528, doi: 10.1109/ROBOT.2001.933002.

[7] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2004, vol. 3, pp. 2149–2154, doi: 10.1109/IROS.2004.1389727.

[8] O. Michel, "Webots: Professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39–42, 2004.

[9] S. Chitta, I. Sucan, and S. Cousins, "Moveit! [Ros Topics]," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 18–19, Mar. 2012, doi: 10.1109/MRA.2011.2181749.

[10] B. Vanderborght, "The impact of hardware and open-source initiatives on robotics [From the Editor's Desk]," *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, p. 4, Mar. 2017, doi: 10.1109/MRA.2016.2645443.

[11] G. Niezen, P. Eslambolchilar, and H. Thimbleby, "Open-source hardware for medical devices," *BMJ Innov.*, vol. 2, no. 2, pp. 78–83, Apr. 2016, doi: 10.1136/bmjinnov-2015-000080.

[12] H. Lim, J. Park, D. Lee, and H. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 33–45, Sep. 2012, doi: 10.1109/MRA.2012.2205629.

[13] J. M. Pearce, *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*. Boston, MA, USA: Newnes, 2013.

[14] R. Heradio et al., "Open-source hardware in education: A systematic mapping study," *IEEE Access*, vol. 6, pp. 72,094–72,103, Nov. 2018, doi: 10.1109/ACCESS.2018.2881929.

[15] E. Vrochidou, M. Manios, G. A. Papakostas, C. N. Aitsidis, and F. Panagiotopoulos, "Open-source robotics: Investigation on existing platforms and their application in education," in *Proc. 26th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, 2018, pp. 1–6, doi: 10.23919/SOFTCOM.2018.8555860.

[16] M. Liarokapis, "An initiative focusing on open-source robot hardware." Open Robot Hardware. Accessed: Aug. 9, 2021. [Online]. Available: http://www.openrobothardware.org/

[17] "2022-2024 new board members." Open Source Hardware Association. Accessed: Nov. 22, 2021. [Online]. Available: https://www.oshwa.org/

[18] T. Duncombe and J. Pearce. "HardwareX." Accessed: Aug. 9, 2021. [Online]. Available: https://www.journals.elsevier.com/hardwarex

[19] D. Berg, K. Niemeyer, and L. Fleischfresser, "[Editorial] open publishing in engineering," *J. Open Eng.*, Aug. 25, 2016, doi: 10.21428/12302. [Online]. Available: https://www.tjoe.org/pub/open-publishing-in-engineering

[20] L. F. Murillo and T. Wenzel, "Welcome to the journal of open hardware," *J. Open Hardware*, vol. 1, no. 1, pp. 1–2, Mar. 2017, doi: 10.5334/joh.3.

[21] A. Dollar, F. Mondada, A. Rodriguez, and G. Metta, "Open-source and widely disseminated robot hardware [From the Guest Editors]," *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 30–31, Mar. 2017, doi: 10.1109/MRA.2016.2646068.

[22] R. R. Ma, J. T. Belter, and A. M. Dollar, "Hybrid deposition manufacturing: Design strategies for multimaterial mechanisms via three-dimensional printing and material deposition," *J. Mech. Robot.*, vol. 7, no. 2, p. 021002, May 2015, doi: 10.1115/1.4029400.

[23] R. Ma and A. Dollar, "Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption," *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 32–40, Mar. 2017, doi: 10.1109/MRA.2016.2639034.

[24] M. Wüthrich et al., "Trifinger: An open-source robot for learning dexterity," 2021, *arXiv:2008.03596*.

[25] L. Gerez, G. Gao, A. Dwivedi, and M. Liarokapis, "A hybrid, wearable exoskeleton glove equipped with variable stiffness joints, abduction capabilities, and a telescopic thumb," *IEEE Access*, vol. 8, pp. 173,345–173,358, Sep. 2020, doi: 10.1109/ACCESS.2020.3025273.

[26] M.-H. Frouin. "Niryo One documentation." Niryo. Accessed: Jun. 15, 2022. [Online]. Available: https://niryo.com/docs/niryo-one/

[27] "Yale OpenHand project documentation." Yale. Accessed: May 31, 2022. [Online]. Available: https://www.eng.yale.edu/grablab/openhand/

[28] G. Franchi, A. ten Pas, R. Platt, and S. Panzieri, "The Baxter Easyhand: A robot hand that costs $150 us in parts," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 2917–2922, doi: 10.1109/IROS.2015.7353779.

[29] M. V. Liarokapis, A. G. Zisimatos, C. I. Mavrogiannis, and K. J. Kyriakopoulos, "OpenBionics: An open-source initiative for the creation of affordable, modular, light-weight, underactuated robot hands and prosthetic devices," in *Proc. 2nd ASU Rehabil. Robot. Workshop*, Feb. 2014.

[30] A. G. Zisimatos, M. V. Liarokapis, C. I. Mavrogiannis, and K. J. Kyriakopoulos, "Open-source, affordable, modular, light-weight, underactuated robot hands," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2014, pp. 3207–3212, doi: 10.1109/IROS.2014.6943007.

[31] M. Ahn et al., "ROBEL: Robotics benchmarks for learning with low-cost robots," in *Proc. Conf. Robot Learn. (CoRL)*, 2019, pp. 1300–1333.

[32] H. Park and D. Kim, "An open-source anthropomorphic robot hand system: HRI hand," *HardwareX*, vol. 7, p. e00100, Apr. 2020, doi: 10.1016/j.ohx.2020.e00100.

[33] G. P. Kontoudis, M. Liarokapis, and K. G. Vamvoudakis, "An adaptive, humanlike robot hand with selective interdigitation: Towards robust grasping and dexterous, in-hand manipulation," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots (Humanoids)*, 2019, pp. 251–258, doi: 10.1109/Humanoids43949.2019.9035037.

[34] G. P. Kontoudis, M. V. Liarokapis, A. G. Zisimatos, C. I. Mavrogiannis, and K. J. Kyriakopoulos, "Open-source, anthropomorphic, underactuated robot hands with a selectively lockable differential mechanism: Towards affordable prostheses," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 5857–5862, doi: 10.1109/IROS.2015.7354209.

[35] N. E. Krausz, R. A. L. Rorrer, and R. F. f Weir, "Design and fabrication of a six degree-of-freedom open source hand," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 5, pp. 562–572, May 2016, doi: 10.1109/TNSRE.2015.2440177.

[36] T. Mańkowski, J. Tomczyński, K. Walas, and D. Belter, "Put-hand—Hybrid industrial and biomimetic gripper for elastic object manipulation," *Electronics*, vol. 9, no. 7, 2020, Art. no. 1147, doi: 10.3390/electronics9071147.

[37] "Brunel hand 2.0." Open Bionics. Accessed: Aug. 9, 2021. [Online]. Available: https://openbionicslabs.com/shop/brunel-hand

[38] K. Berce. "3D printed exoskeleton arm." Hackaday. Accessed: May 31, 2022. [Online]. Available: https://hackaday.io/project/166841-3d-printed-exoskeleton-arm

[39] L. Gerez, J. Chen, and M. Liarokapis, "On the development of adaptive, tendon-driven, wearable Exo-gloves for grasping capabilities enhancement," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 422–429, Apr. 2019, doi: 10.1109/LRA.2019.2890853.

[40] T. A. Georgiou, D. Asnaghi, A. Liang, and A. M. Agogino, "The Sparthan three-dimensional printed Exo-glove: A preliminary evaluation of performance via case study," *J. Med. Devices*, vol. 13, no. 3, 2019, Art. no. 031007, doi: 10.1115/1.4043976.

[41] A. A. Portnova, G. Mukherjee, K. M. Peters, A. Yamane, and K. M. Steele, "Design of a 3D-printed, open-source wrist-driven orthosis for individuals with spinal cord injury," *PLoS One*, vol. 13, no. 2, Feb. 2018, Art. no. e0193106, doi: 10.1371/journal.pone.0193106.

[42] A. I. McPherson, V. V. Patel, P. R. Downey, A. A. Alvi, M. E. Abbott, and H. S. Stuart, "Motor-augmented wrist-driven orthosis: Flexible grasp assistance for people with spinal cord injury," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2020, pp. 4936–4940, doi: 10.1109/EMBC44109.2020.9176037.

[43] R. Freire, B. R. Glowacki, R. R. Williams, M. Wonnacott, A. Jamieson-Binnie, and D. R. Glowacki, "OMG-VR: Opensource mudra gloves for manipulating molecular simulations in VR," 2019, *arXiv:1901.03532*.

[44] "LucidGloves: VR Haptic Gloves on a budget." GitHub. Accessed: May 31, 2022. [Online]. Available: https://github.com/LucidVR/lucidgloves

[45] A. C. Chacin, T. Oozu, and H. Iwata, "IrukaTact: Submersible haptic search glove," in *Proc. 10th Int. Conf. Tangible, Embedded, Embodied Interact. (TEI)*, 2016, pp. 392–397, doi: 10.1145/2839462.2856546.

[46] J. Kilian, A. Neugebauer, L. Scherffig, and S. Wahl, "The unfolding space glove: A wearable spatio-visual to haptic sensory substitution device for blind people," *Sensors*, vol. 22, no. 5, Feb. 2022, Art. no. 1859, doi: 10.3390/s22051859.

[47] Á. L. Muro. "Thor: OpenSource 3D printable robotic arm." Hackaday. Accessed: Aug. 9, 2021. [Online]. Available: https://hackaday.io/project/12989-thor

[48] P. Crnjak. "CM6 - Compliant 3D printed robotic arm." Hackaday. Accessed: May 31, 2022. [Online]. Available: https://hackaday.io/project/180588-cm6-compliant-3d-printed- robotic-arm

[49] "OpenManipulator-X," Robotis, Lake Forest, CA, USA, 2021. Accessed: Aug. 9, 2021. [Online]. Available: https://www.robotis.us/openmanipulator-x/

[50] A. Tai, M. Chun, Y. Gan, M. Selamet, and H. Lipson, "Para: A one-meter reach, two-kg payload, three-DOF open source robotic arm with customizable end effector," *HardwareX*, vol. 10, 2021, Art. no. e00209, doi: 10.1016/j.ohx.2021.e00209.

[51] "Dexter: An open-source, 3D printed, high precision robotic arm with trainability." GitHub. Accessed: Jun. 15, 2022. [Online]. Available: https://github.com/HaddingtonDynamics/Dexter/wiki/Hardware

[52] H. Admoni et al. "MyKeepon project." Accessed: Aug. 9, 2021. [Online]. Available: https://scazlab.yale.edu/sites/default/files/files/admoni_thesis.pdf

[53] M. Suguitan and G. Hoffman, "Blossom: A handcrafted open-source robot," *J. Hum.-Robot Interact.*, vol. 8, no. 1, pp. 1–27, Mar. 2019, doi: 10.1145/3310356.

[54] M. Lapeyre et al., "Poppy project: Open-source fabrication of 3D printed humanoid robot for science, education and art," in *Proc. Digit. Intell.*, Nantes, France, Sep. 2014, p. 6.

[55] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: An open platform for research in embodied cognition," in *Proc. 8th Workshop Perf. Metrics Intell. Syst.*, New York, NY, USA: Association for Computing Machinery, 2008, pp. 50–56.

[56] C. Vandevelde and J. Saldien, "An open platform for the design of social robot embodiments for face-to-face communication," in *Proc. 11th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, 2016, pp. 287–294, doi: 10.1109/HRI.2016.7451764.

[57] D. Casas-Bocanegra et al., "An open-source social robot based on compliant soft robotics for therapy with children with ASD," *Actuators*, vol. 9, no. 3, Sep. 2020, Art. no. 91, doi: 10.3390/act9030091.

[58] "CASTOR robot documentation." GitHub. Accessed: May 31, 2022. [Online]. Available: https://github.com/CastorProject/CASTOR_Robot/wiki

[59] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong, "Development of open humanoid platform DARwIn-OP," in *Proc. SICE Annu. Conf.*, 2011, pp. 2178–2181.

[60] M. Schwarz, M. Schreiber, S. Schüller, M. Missura, and S. Behnke, "NimbRo-OP humanoid teensize open platform," in *Proc. IEEE-RAS 7th Int. Conf. Humanoid Robots Workshop Humanoid Soccer Robots*, Nov. 2012, pp. 1–6.

[61] G. Ficht et al., "NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)*, 2018, pp. 1–9, doi: 10.1109/HUMANOIDS.2018.8625038.

[62] G. Langevin. "InMoov-open source 3D printed life-size robot." Inmoov. Accessed: Aug. 9, 2021. [Online]. Available: http://inmoov.fr

[63] "Poppy project forum." Poppy. Accessed: May 31, 2022. [Online]. Available: https://poppy.discourse.group/

[64] "InMoov robot community." InMoov. Accessed: May 31, 2022. [Online]. Available: http://inmoov.fr/community-v2/

[65] B. Johansson, T. A. Tjøstheim, and C. Balkenius, "Epi: An open humanoid platform for developmental robotics," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 2, 2020, Art. no. 1729881420911498, doi: 10.1177/1729881420911498.

[66] M. Kerzel, E. Strahl, S. Magg, N. Navarro-Guerrero, S. Heinrich, and S. Wermter, "Nico — Neuro-inspired companion: A developmental humanoid robot platform for multimodal interaction," in *Proc. 26th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, 2017, pp. 113–120, doi: 10.1109/ROMAN.2017.8172289.

[67] Z. Faraj et al., "Facially expressive humanoid robotic face," *HardwareX*, vol. 9, Apr. 2021, Art. no. e00117, doi: 10.1016/j.ohx.2020.e00117.

[68] L. Natale, C. Bartolozzi, F. Nori, G. Sandini, and G. Metta, "iCub," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds. Dordrecht, The Netherlands: Springer Nature Switzerland AG, 2019, pp. 291–323.

[69] F. Grimminger et al., "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3650–3657, Apr. 2020, doi: 10.1109/LRA.2020.2976639.

[70] A. T. Spröwitz et al., "Oncilla robot: A versatile open-source quadruped research robot with compliant pantograph legs," *Frontiers Robot AI*, vol. 5, Jun. 2018, Art. no. 67, doi: 10.3389/frobt.2018.00067.

[71] J. Ramos, Y. Ding, Y. Woo Sim, K. Murphy, and D. Block, "Hoppy: An open-source and low-cost kit for dynamic robotics education," 2020, *arXiv:2010.14580.*

[72] A. F. Azocar, L. M. Mooney, L. J. Hargrove, and E. J. Rouse, "Design and characterization of an open-source robotic leg prosthesis," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics (Biorob)*, 2018, pp. 111–118, doi: 10.1109/BIOROB.2018.8488057.

[73] M. Hutter et al., "ANYmal - A highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 38–44, doi: 10.1109/IROS.2016.7758092.

[74] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the MIT cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 509–522, Jun. 2017, doi: 10.1109/TRO.2016.2640183.

[75] N. Kau, A. Schultz, N. Ferrante, and P. Slade, "Stanford doggo: An open-source, quasi-direct-drive quadruped," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, 2019, pp. 6309–6315, doi: 10.1109/ICRA.2019.8794436.

[76] N. Kau, A. Schultz, T. Punnoose, L. Lee, and Z. Manchester, "Woofer and Pupper: Low-cost open-source quadrupeds for research and education," 2020. [Online]. Available: https://www.seas.upenn.edu/~posa/DynamicWalking2020/695-1037-1-RV.pdf

[77] "SpotMicroAI." Accessed: May 31, 2022. [Online]. Available: https://spotmicroai.readthedocs.io/en/latest/

[78] M. Rahme and A. Elarabawy. "OpenQuadruped." GitHub. Accessed: May 31, 2022. [Online]. Available: https://github.com/OpenQuadruped/spot_mini_mini/tree/spot/spot_real

[79] M. Rahme, I. Abraham, M. Elwin, and T. Murphey, "Spotminimini: Pybullet gym environment for gait modulation with Bezier curves, version 2.1.0," 2020. [Online]. Available: https://github.com/moribots/spot_mini_mini

[80] G. Passault, Q. Rouxel, F. Petit, and O. Ly, "METABOT: A low-cost legged robotics platform for education," in *Proc. Int. Conf. Auton. Robot Syst. Competitions (ICARSC)*, 2016, pp. 283–287, doi: 10.1109/ICARSC.2016.52.

[81] S. Lohmann, J. Yosinski, E. Gold, J. Clune, J. Blum, and H. Lipson, "ARACNA: An open-source quadruped platform for evolutionary robotics," in *Proc. 13th Int. Conf. Synthesis Simul. Living Syst. (ALIFE)*, Jul. 2012, pp. 387–392, doi: 10.7551/978-0-262-31050-5-ch051.

[82] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, "Self-modifying morphology experiments with DYRET: Dynamic robot for embodied testing," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, 2019, pp. 9446–9452, doi: 10.1109/ICRA.2019.8793663.

[83] L. Wang, Y. Yang, G. Correa, K. Karydis, and R. S. Fearing, "OpenRoACH: A durable open-source hexapedal platform with onboard robot operating system (ROS)," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, 2019, pp. 9466–9472, doi: 10.1109/ICRA.2019.8794042.

[84] I. Fitzner, Y. Sun, V. Sachdeva, and S. Revzen, "Rapidly prototyping robots: Using plates and reinforced flexures," *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 41–47, Mar. 2017, doi: 10.1109/MRA.2016.2639058.

[85] "Open-source leg (Humotech) forum." Humotech. Accessed: May 31, 2022. [Online]. Available: https://humotech.com/community/

[86] "How to assemble the Ankle: Open-source leg v1.0." Open-Source Leg. Accessed: May 31, 2022. [Online]. Available: https://opensourceleg.com/osl-v2-ankle-assembly/

[87] J. Tamez-Duque, G. Santos, O. Gutiérrez, and R. Soto. "ALICE pediatric exoskeleton." INDI. Accessed: May 31, 2022. [Online]. Available: http://www.indi.global/aliceose

[88] F. Mondada et al., "The e-puck, a robot designed for education in engineering," in *Proc. 9th Conf. Auton. Robot Syst. Competitions*, 2009, vol. 1, no. 1, pp. 59–65.

[89] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, and B. Lennox, "Mona: An affordable open-source mobile robot for education and research," *J. Intell. Robot. Syst.*, vol. 94, nos. 3–4, pp. 761–775, 2019, doi: 10.1007/s10846-018-0866-9.

[90] J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, and M. Abderrahim, "A new open source 3D-printable mobile robotic platform for education," in *Proc. Adv. Auton. Mini Robots*, Berlin, Heidelberg: Springer-Verlag, 2012, pp. 49–62, doi: 10.1007/978-3-642-27482-4_8.

[91] "Zumy." Wiki. Accessed: Aug. 9, 2021. [Online]. Available: https://wiki.eecs.berkeley.edu/biomimetics/Main/Zumy/

[92] A. Nechypurenko, M. Parkachov, and N. Nikolaev. "Veter - Robotics vehicle for researchers and makers." Veter. Accessed: Aug. 9, 2021. [Online]. Available: https://veterobot.com/

[93] R. Bonghim. "NanoSaur." Accessed: Aug. 9, 2021. [Online]. Available: https://nanosaur.ai/

[94] "ESA ExoMy," *ESA Planetary Robot. Lab.*, 2021. Accessed: Aug. 9, 2021. [Online]. Available: https://esa-prl.github.io/ExoMy/

[95] "JPL open source rover." GitHub. Accessed: Aug. 9, 2021. [Online]. Available: https://github.com/nasa-jpl/open-source-rover

[96] T. Foote and M. Wise. "What is a TurtleBot?" TurtleBot Accessed: Aug. 9, 2021. [Online]. Available: https://www.turtlebot.com/

[97] "MIT Racecar." Accessed: Aug. 9, 2021. [Online]. Available: https://mit-racecar.github.io/

[98] S. S. Srinivasa et al., "MuSHR: A low-cost, open-source robotic racecar for education and research," 2019, *arXiv:1908.08031.*

[99] R. Deimel and O. Brock, "A compliant hand based on a novel pneumatic actuator," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2013, pp. 2047–2053, doi: 10.1109/ICRA.2013.6630851.

[100] "PneuFlex tutorial." Robotics. Accessed: May 31, 2022. [Online]. Available: https://www.robotics.tu-berlin.de/menue/software_tutorials/pneuflex_tutorial/

[101] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 161–185, 2016, doi: 10.1177/0278364915592961.

[102] D. P. Holland, E. J. Park, P. Polygerinos, G. J. Bennett, and C. J. Walsh, "The soft robotics toolkit: Shared resources for research and design," *Soft Robot.*, vol. 1, no. 3, pp. 224–230, Sep. 2014, doi: 10.1089/soro.2014.0010.

[103] C. Della Santina et al., “The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots,” *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 48–56, Mar. 2017, doi: 10.1109/MRA.2016.2636366.

[104] V. Zykov, P. Williams, N. Lassabe, and H. Lipson, “Molecubes extended: Diversifying capabilities of open-source modular robotics,” in *Proc. IEEE Int. Conf. Intell. Robots Syst. Self-Reconfigurable Robot. Workshop*, 2008, pp. 22–26.

[105] V. Zykov, A. Chan, and H. Lipson, “Molecubes: An open-source modular robotics kit,” in *Proc. Self-Reconfigurable Robot. Workshop (IROS)*, 2007, pp. 3–6.

[106] H. Woern, M. Szymanski, and J. Seyfried, “The I-SWARM project,” in *Proc. 15th IEEE Int. Symp. Robot Hum. Interact. Commun. (ROMAN)*, 2006, pp. 492–496, doi: 10.1109/ROMAN.2006.314376.

[107] M. Rubenstein, C. Ahler, and R. Nagpal, “Kilobot: A low cost scalable robot system for collective behaviors,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2012, pp. 3293–3298, doi: 10.1109/ICRA.2012.6224638.

[108] M. Allwright, W. Zhu, and M. Dorigo, “An open-source multi-robot construction system,” *HardwareX*, vol. 5, Apr. 2019, Art. no. e00050, doi: 10.1016/j.ohx.2018.e00050.

[109] “Arduino.” Accessed: Aug. 9, 2021. [Online]. Available: https://www.arduino.cc/

[110] “Raspberry Pi Foundation.” Accessed: Oct. 20, 2021. [Online]. Available: https://www.raspberrypi.org/

[111] Y. Tenzer, L. P. Jentoft, and R. D. Howe, “The feel of MEMS barometers: Inexpensive and easily customized tactile array sensors,” *IEEE Robot. Autom. Mag.*, vol. 21, no. 3, pp. 89–95, Sep. 2014, doi: 10.1109/MRA.2014.2310152.

[112] M. Lambeta et al., “DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3838–3845, Jun. 2020, doi: 10.1109/LRA.2020.2977257.

[113] S. Joonhigh, N. Kuppuswamy, A. Beaulieu, A. Alspach, and R. Tedrake, “Soft bubble grippers for robust and perceptive manipulation,” Toyota Res. Inst., Los Altos, CA, USA, 2021. Accessed: Aug. 9, 2021. [Online]. Available: https://punyo.tech/

[114] “Open Electronics.” Accessed: Oct. 20, 2021. [Online]. Available: https://www.open-electronics.org/

[115] “Kitspace.” Accessed: Oct. 20, 2021. [Online]. Available: https://kitspace.org/

[116] “Hackaday.” Accessed: Oct. 20, 2021. [Online]. Available: https://hackaday.io/

[117] G. Pandey and A. Vora, “Open electronics for medical devices: State-of-art and unique advantages,” *Electronics*, vol. 8, no. 11, Nov. 2019, Art. no. 1256, doi: 10.3390/electronics8111256.

[118] T. D. Ngo, “Open-source electronics platforms: Development and applications,” *Electronics*, vol. 8, no. 4, Apr. 2019, Art. no. 428, doi: 10.3390/electronics8040428.

[119] “Piccolo.” Accessed: Aug. 9, 2021. [Online]. Available: http://piccolo.cc/

[120] “FarmBot.” Accessed: Aug. 9, 2021. [Online]. Available: https://farm.bot/

[121] M. O. Martinez et al., “3-D printed haptic devices for educational applications,” in *Proc. IEEE Haptics Symp. (HAPTICS)*, 2016, pp. 126–133, doi: 10.1109/HAPTICS.2016.7463166.

[122] J. Forsslund, M. Yip, and E.-L. Sallnäs, “WoodenHaptics: A starting kit for crafting force-reflecting spatial haptic devices,” in *Proc. 19th Int. Conf. Tangible, Embedded, Embodied Interact.*, Stanford, CA, USA: Association for Computing Machinery, 2015, pp. 133–140, doi: 10.1145/2677199.2680595.

[123] A. Eilering, G. Franchi, and K. Hauser, “ROBOPuppet: Low-cost, 3D printed miniatures for teleoperating full-size robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1248–1254, doi: 10.1109/IROS.2014.6942717.

[124] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da Vinci® surgical system,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2014, pp. 6434–6439, doi: 10.1109/ICRA.2014.6907809.

[125] B. Hannaford et al., “Raven-II: An open platform for surgical robotics research,” *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 954–959, Apr. 2013, doi: 10.1109/TBME.2012.2228858.

[126] V. Patel, S. Krishnan, A. Goncalves, and K. Goldberg, “SPRK: A low-cost Stewart platform for motion study in surgical robotics,” in *Proc. Int. Symp. Med. Robot. (ISMR)*, 2018, pp. 1–6, doi: 10.1109/ISMR.2018.8333300.

[127] A. B. Clark, V. Mathivannan, and N. Rojas, “A continuum manipulator for open-source surgical robotics research and shared development,” *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 1, pp. 277–280, Feb. 2021, doi: 10.1109/TMRB.2020.3041391.

[128] M. Beccani et al., “Systematic design of medical capsule robots,” *IEEE Des. Test*, vol. 32, no. 5, pp. 98–108, Oct. 2015, doi: 10.1109/MDAT.2015.2459591.

[129] G. Gorjup and M. Liarokapis, “A low-cost, open-source, robotic airship for education and research,” *IEEE Access*, vol. 8, pp. 70,713–70,721, Apr. 2020, doi: 10.1109/ACCESS.2020.2986772.

[130] N. Shahrubudin, T. C. Lee, and R. Ramlan, “An overview on 3D printing technology: Technological, materials, and applications,” *Procedia Manuf.*, vol. 35, pp. 1286–1296, Aug. 2019, doi: 10.1016/j.promfg.2019.06.089.

[131] S. Das, D. L. Bourell, and S. Babu, “Metallic materials for 3D printing,” *MRS Bull.*, vol. 41, no. 10, pp. 729–741, Oct. 2016, doi: 10.1557/mrs.2016.217.

[132] B. Gorissen, D. Reynaerts, S. Konishi, K. Yoshida, J.-W. Kim, and M. De Volder, “Elastic inflatable actuators for soft robotic applications,” *Adv. Mater.*, vol. 29, no. 43, 2017, Art. no. 1604977, doi: 10.1002/adma.201604977.

[133] R. Merz, F. Prinz, K. Ramaswami, M. Terk, and L. Weiss, “Shape deposition manufacturing,” in *Proc. Int. Solid Freeform Fabr. Symp.*, 1994, pp.1–8.

[134] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015, doi: 10.1038/nature14543.

[135] C. E. Okwudire and H. V. Madhyastha, “Distributed manufacturing for and by the masses,” *Science*, vol. 372, no. 6540, pp. 341–342, Apr. 2021, doi: 10.1126/science.abg4924.

[136] G. P. Kontoudis, M. V. Liarokapis, A. G. Zisimatos, C. I. Mavrogiannis, and K. J. Kyriakopoulos, “How to create affordable, anthropomorphic, personalized, light-weight prosthetic hands,” 2015. [Online]. Available: https://openbionics.org/2015_TR_OpenBionicsProstheticHandsGuide.pdf

[137] “Piccolo project bill of materials.” GitHub. Accessed: May 31, 2022. [Online]. Available: https://github.com/DiatomStudio/Piccolo/wiki/Piccolo-Bill-of-Materials

[138] “PUNYO bubble gripper bill of materials (BOM) v1.0.” PUNYO. Accessed: May 31, 2022. [Online]. Available: https://punyo.tech/downloads/Bubble_Gripper_BOM_v1.0.pdf

[139] “Poppy project documentation.” Poppy. Accessed: May 31, 2022. [Online]. Available: https://docs.poppy-project.org/en/

[140] “WoodenHaptics project documentation.” WoodenHaptics. Accessed: May 31, 2022. [Online]. Available: https://woodenhaptics.org/build.html

[141] “Soft robotics toolkit – Actuator molding documentation.” Soft Robotics Toolkit. Accessed: May 31, 2022. [Online]. Available: https://softroboticstoolkit.com/book/fr-mold-actuator-body

[142] “ESA-PRL ExoMy rover 3D printing.” GitHub. Accessed: May 31, 2022. [Online]. Available: https://github.com/esa-prl/ExoMy/wiki/3D-Printing

[143] “PUNYO bubble gripper bill of materials (BOM) v1.0.” PUNYO. Accessed: May 31, 2022. [Online]. Available: https://punyo.tech/downloads/Bubble Gripper_Build_Instructions_v1.0.pdf

[144] “Choose a license appendix.” Choose-A-License. Accessed: May 31, 2022. [Online]. Available: https://choosealicense.com/appendix/

[145] “About CC licenses,” Creative Commons, Mountain View, CA, USA, 2022. Accessed: May 31, 2022. [Online]. Available: https://creativecommons.org/about/cclicenses/

[146] “Creative commons licenses,” Creative Commons, Mountain View, CA, USA, 2022. Accessed: Oct. 31, 2022. [Online]. Available: https://libguides.gwu.edu/opentextbooks/creative_commons

[147] A. Hansen and T. J. Howard, “The current state of open source hardware: The need for an open source development platform,” in *ICoRD'13*, A. Chakrabarti and R. Prakash, Eds. New Delhi, India: Springer Nature, 2013, pp. 977–988.

[148] “GNU general public license (GPL) v3.” GNU Operating System. Accessed: May 31, 2022. [Online]. Available: http://www.gnu.org/licenses/gpl-3.0.html

[149] “MIT license.” Open Source Initiative. Accessed: May 31, 2022. [Online]. Available: https://opensource.org/licenses/MIT

[150] “The 3-clause BSD license.” Open Source Initiative. Accessed: May 31, 2022. [Online]. Available: https://opensource.org/licenses/BSD-3-Clause

[151] “Apache licenses.” Apache. Accessed: May 31, 2022. [Online]. Available: https://www.apache.org/licenses/

[152] “TAPR open hardware license.” TAPR. Accessed: May 31, 2022. [Online]. Available: https://tapr.org/the-tapr-open-hardware-license/

[153] “CERN open hardware licence.” Open Hardware Repository. Accessed: May 31, 2022. [Online]. Available: https://ohwr.org/project/cernohl/wikis/home

[154] “Solderpad Hardware License.” Accessed: May 31, 2022. [Online]. Available: http://solderpad.org/licenses/

[155] K. J. Stewart, A. P. Ammeter, and L. M. Maruping, “Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects,” *Inf. Syst. Res.*, vol. 17, no. 2, pp. 126–144, Jun. 2006, doi: 10.1287/isre.1060.0082.

[156] T. O'Reilly, “Lessons from open-source software development,” *Commun. ACM*, vol. 42, no. 4, pp. 32–37, Apr. 1999, doi: 10.1145/299157.299164.

[157] G. Von Krogh and S. Spaeth, “The open source software phenomenon: Characteristics that promote research,” *J. Strategic Inf. Syst.*, vol. 16, no. 3, pp. 236–253, Sep. 2007, doi: 10.1016/j.jsis.2007.06.001.

[158] D. Wu, D. W. Rosen, L. Wang, and D. Schaefer, “Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation,” *Comput.-Aided Des.*, vol. 59, pp. 1–14, Feb. 2015, doi: 10.1016/j.cad.2014.07.006.