Roman: Making Everyday Objects Robotically Manipulable with 3D-Printable Add-on Mechanisms

Jiahao Li
UCLA HCI Research
ljhnick@g.ucla.edu

Alexis A Samoylov
UCLA HCI Research
alexiy.samoylov@gmail.com

Jeeeun Kim
HCIED Lab, Texas A&M University
jeeeun.kim@tamu.edu

Xiang 'Anthony' Chen
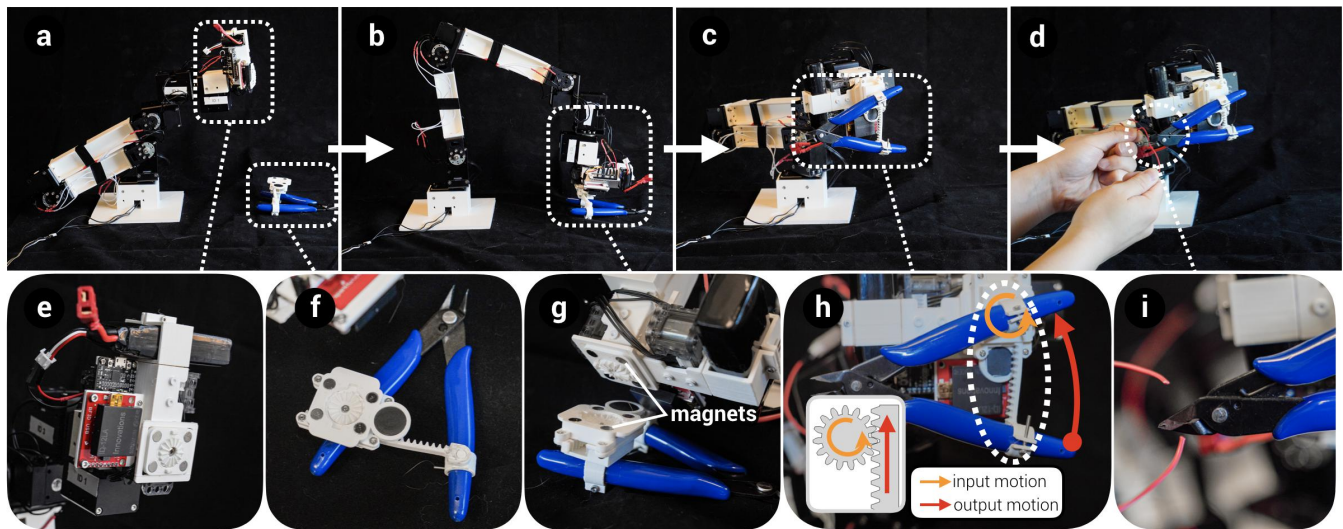UCLA HCI Research
xac@ucla.edu

Figure 1: Roman is a novel robotic design making everyday handheld objects more robotically manipulable, *i.e.*, easier to be manipulated by a conventional robotic arm. This figure shows a sequence of a Roman-enabled robotic arm picking up a wire cutter on the desk and performing a wire cutting task collaborating with a human (a-d). Roman provides a magnetic gripper (e) for the robotic arm to easily attach and augment the wire cutter with Roman mechanism (f). Snapping in the mechanism using magnets (g), the gripper can actuate the gear-rack movement on the wire cutter (h) to perform the cutting task by squeezing the cutter's handles (i).

ABSTRACT

One important vision of robotics is to provide physical assistance by manipulating different everyday objects, e.g., hand tools, kitchen utensils. However, many objects designed for dexterous hand-control are not easily manipulable by a single robotic arm with a generic parallel gripper. Complementary to existing research on developing grippers and control algorithms, we present Roman, a suite of hardware design and software tool support for robotic engineers to create 3D printable mechanisms attached to everyday handheld objects, making them easier to be manipulated by conventional robotic arms. The Roman hardware comes with a versatile magnetic gripper that can snap on/off handheld objects and drive add-on mechanisms to perform tasks. Roman also provides software support to register and author control programs. To validate our approach, we designed and fabricated Roman mechanisms for 14 everyday objects/tasks presented within a design space and conducted expert interviews with robotic engineers indicating that Roman serves as a practical alternative for enabling robotic manipulation of everyday objects.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Robotic grasping and manipulation; handheld objects augmentation; mechanism design.

1 INTRODUCTION

Various types of robots that inhabit our living spaces, such as vacuum cleaners, robotic toys, home/office-assistive robots, promise to aid humans with numerous everyday tasks. Future robots are expected to provide physical assistance for the elderly at home or help with household chores that require the use of diverse tools and everyday objects [26].

Many recent advances in robotic research (primarily on manipulation [17, 38, 53, 54]) aim to solve robotic arms' manipulation problem for niche, fixed tasks, *e.g.*, opening medicine bottles, handling cookware, and sorting waste. However, creating a *universally dexterous* robotic arm remains challenging as there are a large number of everyday objects that are not robotically manipulable, *i.e.*, difficult to manipulate by a consumer-level robotic arm with a generic parallel gripper, especially so for those *dynamic* objects that have multiple mechanically movable parts, *e.g.*, a pair of pliers and a spray bottle with a pump pushable toward the body.

To tackle this robotic manipulability challenge, prior work often considers the action of grippers and manipulators as completely decoupled, reducing the complexity of task planning and control involved [3]. To date, researchers have focused on developing manipulation strategies by analyzing the best grasping points of *static* objects [38, 53]. However, when it comes to *dynamic* objects such as a spray bottle with a pump pushable toward the body, prior work tends to mimic the pose and force of a human with *multiple* robotic arm [13, 37] or dexterous robotic hand [1] rather than tackling the task self-containedly with a *single* robotic arm. Meanwhile, on the objects' side, augmenting objects with actuable mechanisms is a new approach to enhance their interactivity or functionality (*e.g.*, [8, 33]); however, little has been explored on how to enable robotic arms to better manipulate such augmented objects.

We present Roman—a suite of hardware design and software tool for robotic engineers to make everyday handheld objects more Robotically manipulable by a consumer-grade 6-DoF robotic arm.

Roman's hardware components consist of *(i)* a library of 3D printable powerless mechanisms that are attached to and can drive different handheld objects to perform specific tasks, *e.g.*, squeezing a cutter to cut wires (Figure 1a→d); *(ii)* a gripper (Figure 1e) that uses magnets to securely and automatically attach/detach a robotic arm to/from an object[1]. The gripper also contains an RFID-based module to recognize which object the robotic arm is expected to manipulate thus to run the corresponding control program for specific tasks.

Roman's software component is a user interface for robotic engineers to register and specify the motor input of custom tasks with pre-defined templates and real-time feedback. A user can rapidly and iteratively author a control program for a Roman-enabled robotic arm to manipulate a handheld object. Note that Roman does *not* reinvent tools for generating 3D models of add-on mechanism, which is already supported (*e.g.*, [9, 33]); rather, Roman's software focuses on creating a control program to operate such mechanisms as different handheld objects require unique sequences of action to perform a specific task, which would otherwise be tedious to specify even for robotic experts.

To validate Roman, we first designed and fabricated mechanisms for 14 everyday objects to demonstrate how a generic 6-DoF robotic arm (built with 3D printed links and Dynamxiel XH-540 motors[2]) can manipulate these objects to perform specific tasks. Further, we conducted a preliminary interview with four robotic engineers. Participants found Roman's approach complementary to the current research on robotic manipulation within the human environment, which could be beneficial in specific task scenarios such as cooking, electronics assembly, and caring for house plants. Provided with the pre-fabricated gripper and attaching mechanisms, all participants were able to use Roman's software module to replicate the wire cutter scenario (Figure 1).

1.1 Contributions

Overall, Roman empowers robotics engineers to use 3D printable powerless mechanisms attached to different objects for enhancing these objects' robotic manipulability with task-specific control program embedded in the mechanisms. Our specific contributions are as follows.

- **Categorization of the robotic manipulability problem in everyday handheld objects (§2)**, including the challenge of speed and range of motion, and the challenge of dexterity when manipulating an dynamic object with its constituent parts (squeezing, twisting, and pumping);
- **A library of 3D printable add-on mechanisms able to manipulate both static and dynamic objects (§6.2)** based on variations and combinations of spur gears, bevel gears, gear racks, and pin-in-slot mechanisms, which can be attached to everyday objects to enhance their manipulability by a consumer-grade 6-DoF robotic arm;
- **The design of a versatile magnetic gripper (§6.1)** that can securely attach to the add-on mechanisms on an object, run the control program to drive the mechanism, and detach from the mechanism without outside intervention;
- **Software that enables robotic engineers to interactively author a program (§7)** to control the Roman mechanism-installed hardware to perform object-specific tasks.

1.2 Limitation

At present, Roman has not achieved total autonomy, as there is no sensing modules to detect where an object is positioned or how it is oriented for pick-up (which is a separate topic well studied in robotics and computer vision [14, 18]). Currently, in most of our examples, the object is handed off to the robotic arm by a human (*e.g.*, kitchen utensils) or the position of the object is manually input

---

[1]Caveat: currently, an object's placement needs to be known for auto-attachment

[2]https://www.robotis.us/dynamixel-xh540-w150-t/

to the robotic control program by the human user (*e.g.*, the spray bottle example).

2 CATEGORIZING CHALLENGES OF HANDHELD OBJECTS FOR ROBOTIC MANIPULATION

Many everyday objects are not robotically manipulable due to the following challenges:

- **Speed and range of motion challenges** when manipulating an object as a whole. For example, objects that require a high speed/frequency of manipulation may exceed the motor's capability at the end-effector, *e.g.*, a rapidly rotating whisker at a high speed to whip cream (Figure 2a). Meanwhile, manipulating a screwdriver at certain angles might exceed the reach of a situated robotic arm (Figure 2b)
- **Dexterity-related challenges** when manipulating an object by its constituent parts. For example, to cut a wire, a robotic arm would first need to grasp the cutter legs then apply a steady force to squeeze the handles (Figure 2c); however, a generic parallel gripper might find it hard to both securely grasp *and* squeeze the cutter. Bi-manual manipulation is even harder, *e.g.*, opening a jar lid (Figure 2d) is almost impossible for a single robotic arm and requires either a holding stand or assistance from another robotic arm or a human hand.

Our goal is to make everyday **handheld objects** manipulable by a consumer-grade generic 6-DoF robotic arm. To achieve this, we first need to understand *what* objects are currently inaccessible and *how* they are inaccessible for robotic manipulation. Everyday objects are mostly designed to be manipulated by humans. With flexible fingers and the coordination of four limbs, humans are able to manipulate a wide range of objects with different manipulation complexities, from picking up a cup to playing a piano. In contrast, there exist a large number of everyday objects that are not manipulable by generic robotic arms due to the fact that a general-purpose gripper has limitations in performing different types of grasping and manipulation tasks [3]. As shown in Figure 4, we consider two categories of such objects: those manipulated as a whole (*e.g.*, raising/lowering a knife, rotating a screwdriver) *vs.* those manipulated by constituent parts (*e.g.*, squeezing a pair of pliers' handles, twisting a pepper grinder cap against the bottle, pushing the pump towards the body of a bottle of hand sanitizer).

2.1 Objects Manipulated as a Whole (Static Objects)

As shown in Figure 4A-B, objects in this category require a 'grasp & move' type of manipulation as the body of the object moves as a whole while performing tasks, *e.g.*, the whole knife moves vertically while chopping vegetables (linear motion), the whole egg beater rotates when mixing eggs (rotational motion). Although in theory such objects can be manipulated by a robotic arm—by first grasping it using a gripper and then performing manipulation by a series of joint rotations to create the movement, there remain two limitations that prevent the robotic arm from manipulating such objects to perform tasks as well as humans: Speed and Range of motion.

2.1.1 Speed limitation. A robotic arm mostly manipulates objects at its end-effector, which generates large loads on the arm due to the moment of inertia. Therefore, tasks such as knife chopping (Figure 4A1), which require a fast periodic motion at the end-effector may exceed the robotic arm's capability.

2.1.2 Range of motion limitation. Using a conventional robotic gripper, some object manipulation requires a certain type of grasp, *e.g.*, rotating a screwdriver (Figure 3a). However, when the object is further way, even though it is still within reach of the robotic arm, the grasp is different and as such, it is no longer able to afford the same manipulation (Figure 3b). In other words, grasping the tool at a certain angle might render the manipulation impossible because reaching that angle would already constrain the robotic arm's joint rotation, thus limiting how it can perform subsequent manipulation (Figure 3b). In some edge cases, the manipulation might be interfered with by the physical surroundings, *e.g.*, the ground (Figure 3c). In robotic terms, under such circumstances, the robotic arm is said to be outside of its *dexterous space* [35] when having to perform the manipulation at certain angles.

2.2 Objects Manipulated by Constituent Parts (Dynamic Objects)

In contrast to objects manipulated as a whole, there exist objects made up of and manipulated by their movable parts, as shown in Figure 4C-E. Manipulating such objects tends to be more difficult for a robotic arm, which needs to first grasp the object stably and then perform a dexterous manipulation like human hand(s), *e.g.*, grasping a pepper grinder and then twisting the grinding cap (Figure 4D2). We summarize the following three types of manipulation that makes objects not manipulable by generic robotic arms, each of which can be either one- or bi-directional.

2.2.1 Squeezing. In order to manipulate objects that require 'squeezing' (*e.g.*, a wire cutter in Figure 4C2), a robotic arm (*e.g.*, with a common parallel gripper) would have to pick two points on the squeezing handles for a firm grasp, and then apply a steady force to squeeze the object. However, as the best grasping points for performing the squeezing manipulation (*e.g.*, near the tip of the handle) are usually the furthest away from the center of gravity, this makes the grasp unstable and slippery.

2.2.2 Twisting. The twisting force applied to objects *e.g.*, a door knob (Figure 4D1) would produce a rotational torque on the robotic arm itself after securing the grasp, which may cause the whole system to become unstable. This is also the most common failure in the *DARPA Robotic Challenge* [30].

2.2.3 Pumping. Different from the above, after grasping an object, the robotic arm needs an additional contact surface to perform the pumping task (*e.g.*, to press the hand sanitizer in Figure 4E1), which makes it nearly impossible to perform by a common gripper with a parallel design.

Finally, note that some objects (*e.g.*, pepper grinder, pump) would require bi-manual manipulation. In other words, a *single* robotic arm, even as dexterous as a human hand, would find it challenging to manipulate these objects.
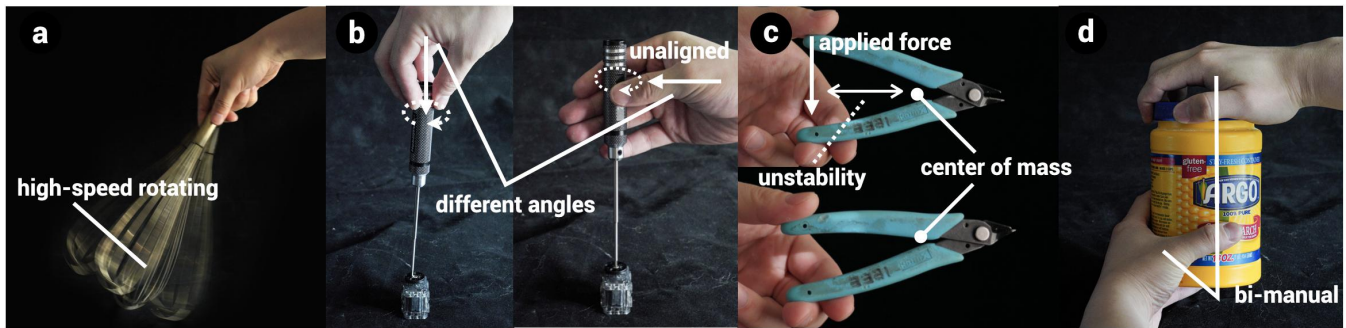
Figure 2: A human hand manipulating a variety of objects that present challenges for robotic arms such as using a whisk (a), rotating a screwdriver (b), squeezing a wire cutter (c), and opening a jar lid (d).
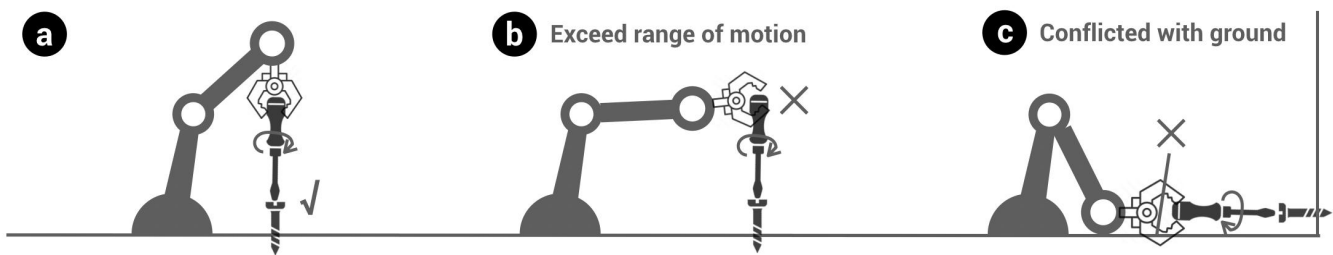

Figure 3: Why manipulating a screwdriver might require range of motion beyond a robotic arm's capability: (a) normal working grasp/distance; (b) when the distance increases, the grasp changes and the end-effector can no longer rotate around the screwdriver's axis; (c) in some edge cases, the manipulation might be interfered by the physical surroundings, *e.g.*, the ground.

Figure 4: Everyday handheld objects are often not manipulable by a generic robotic arm with a common parallel gripper: when manipulating an object as a whole, speed and range of motion are two main limitations and; when manipulating an object by its constituent parts, the dexterity required to both grasp and perform a range of manipulation (squeezing/releasing, twisting and pumping) is the main challenge.

3 RELATED WORK

3.1 Augmenting Generic Robotic Interfaces and Everyday Objects

Roman can be thought of as enabling two types of augmentation: *(i)* augmenting a generic robotic arm so that it can attach to and manipulate various handheld objects and *(ii)* augmenting everyday objects so that they become manipulable by the robotic arm.

3.1.1 Augmenting generic physical interfaces. Existing work has explored various designs for extending the functionality of generic interfaces. For example, HERMITS extended the capability of self-propelled tangible user interfaces (TUIs) by designing mechanical shell add-ons for TUIs to dock and drive dynamic interactions [40]. Katakura *et al.* introduced a novel way of augmenting a 3D printer head into a robotic manipulator with the mechanical attachments printed by the printer itself [24, 25]. Other researchers augmented a pin-based display by the conversion of the mechanical motion of pins with passive mechanisms to enrich the pins' dynamic interaction [41, 50]. HapLinkage simulated the motion and haptic feedback of virtual hand tools using linkage mechanisms and generated a design space of handheld tools based on motion types and force profiles [34]. Davidoff *et al.* designed mechanical actuators based on the Lego MindStorm toolkit to operate on switches that were intended for humans only [12]. In comparison, Roman's mechanisms augment the physical interface of a robotic arm, *i.e.*, the end-effector, with a versatile magnetic gripper with a focus on assisting the robotic arm to manipulate different everyday objects that would otherwise be difficult to manipulate.

3.1.2 Augmenting everyday objects. Another focus of previous research is extending the capability of everyday objects. Reprise makes handheld objects easy-to-manipulate by people with disabilities by generating adaptions attached to the object [8]. Medley enhanced the material properties of 3D printed objects by embedding different reusable materials into the model [7]. Romeo extended the default functionality of everyday objects by embedding a transformable robotic arm into the 3D model of the object [32]. RetroFab augmented the interactivity of physical interface by adding an enclosure consisting of mechanical and electrical components that could automate physical controls [47], which was later extended by Robiot in automating dynamic everyday objects, *e.g.*, adjusting a lamp's joint angle [33].

RetroFab and Robiot employed *active* mechanism (*i.e.*, with motors onboard), which inevitably makes the augmented object bulky and expensive to scale to the numerous everyday objects in the environment. In contrast, Roman utilized a passively actuable mechanism attached to an object with only mechanical components to be manipulated by a generic robotic arm.

Since Roman proposes a different complementary approach for robotic arms to grasp and manipulate everyday objects, below we will review related work in robotics research, focusing on gripper design and manipulation.

3.2 Relationship to Robotic Research

3.2.1 Gripper design. In Robotics, grippers are the most common type of end-effector and an important medium for robots to interact with the real world. The design of robotic grippers has been extensively studied in academia and industry by researchers and practitioners to explore designs of different types of robotic end effectors, *e.g.*, grippers for pick and place, tight grasping, and more. There are two major strategies in designing a gripper: for general purposes or for a specific task. Researchers have explored different types of general purpose robotic grippers including linkage-based parallel mechanism robotic grippers [22, 28] and compliant under-actuated robotic grippers [10, 31, 36, 39]. Meanwhile, exploring how to configure task-specific grippers is an emergent topic that has gained recent attention. Feix *et al.* provided a taxonomy to categorize the potential tasks and the corresponding design features of the robotic gripper [16]. Researchers have designed grippers for different shapes of target objects [42], *e.g.*, for picking objects lying on flat surfaces [2, 4], for assembly tasks [55], or for operating a heavy load [51].

Furthermore, the design of robotic grippers should not only consider the geometry of the objects, but also the interaction with the environment and the kinetostatic properties of the grippers [3]. Different factors are considered in previous research such as dynamic loads [43] and active surfaces after grasping [19, 44]. These are closely related to the tasks conducted by robotic grippers in daily life interacting with everyday tools, which are summarized by the robotic grasping and manipulation challenge [15]. Roman uniquely combines both strategies by implementing a general purpose versatile gripper that could grasp different objects and object-specific mechanisms that could be driven by the gripper to achieve a dynamic manipulation of everyday handheld objects.

Another important gripper-related topic is to create 'tool changers' that adapt to the different shapes and sizes of everyday objects. There are different types of tool changers for a robotic arm: automatic tool changers that are electro-mechanically actuated [20] or passive mechanisms actuated by a host robot [5, 46]. Similar to our method, researchers also proposed the design of mechanical tool for robots to grasp different daily objects with modular two-finger gripper but only focused on the grasping of objects with different sizes [23]. Adding to that, our method could solve a more complex manipulation problem of manipulating tools with movable parts (Figure 4), which is almost unattainable using a two-finger type end effector. Furthermore, since Roman provides customized mechanisms for different objects that can be driven by the same custom gripper, our method provides a less complex method to program the motion of the arm than solutions that incorporate tool changers.

3.2.2 Manipulation. Besides innovating the gripper design, researchers have also investigated methods of control and task planning that program a given robotic arm to perform tasks in human environment. Some research focused on developing a control strategy based on perception such as analyzing the geometry of the objects to obtain the best grasping point [38, 53] or imitating human operation to open medicine bottles [13, 37]. Some developed new algorithms and system designs for specific contexts such as grasping flat objects [48]. At the application level, robots can assist people with daily living tasks which range from fetching a mug [17] to taking medications [29]. In contrast to the above approaches of enhancing perception and control, Roman aims for a different and complementary goal of achieving manipulation by augmenting everyday objects to be more manipulable by a generic robotic arm.

Different from traditional robotic research, a concept of *dexterous manipulation* was first defined by Okamura *et al.* in which a robotic gripper moves objects from one configuration to another [45], *e.g.*, adjusting the angle of a phone in the hand. Such a concept is still being actively studied by researchers in scenarios such as robotic in-hand manipulation [1, 52, 56]. However, such manipulation requires precise control of the forces and motions of fingered or specialized robotic hands and therefore cannot be accomplished by conventional robotic grippers. Roman, as a complementary solution to dexterous manipulation, enables a conventional robotic arm to connect with a versatile magnetic gripper to grasp and manipulate everyday objects with dexterity enabled by object-specific mechanisms.

4 EXAMPLES: ROMAN MAKES HANDHELD OBJECTS ROBOTICALLY MANIPULABLE

We showcase a series of examples where Roman mechanisms attached to handheld objects make them more manipulable by a consumer-grade 6-DoF robotic arm (equipped with a Roman gripper). There are two major types of application scenarios where the objects need to be robotically manipulable: *(i)* A human collaborates with a Roman-equipped robotic arm, *e.g.*, the human would hand an object over to the robotic arm where it would then be picked up and manipulated, in tasks that the robot is expected to manipulate a lot of different handheld objects, *e.g.*, making a scrambled egg (Figure 6de, Figure 14, etc) and *(ii)* a Roman-equipped robotic arm takes the place of human in performing repetitive tasks by manipulating a hard-to-manipulate object, *e.g.*, a spray bottle. We focus on demonstrating the wide capabilities of Roman in enabling object manipulation, while discussing the technical details of the gripper design and the mechanism generation method later in subsequent sections.

4.1 Manipulating Objects as a Whole

4.1.1 Increasing speed. As mentioned above, some objects may require a high speed of manipulation that exceeds the motor's capability at the end-effector. Figure 5 and Figure 6 showcase several different examples where Roman enables a high speed of manipulation: by attaching a pin-in-slot mechanism, the Roman gripper can drive a knife to perform repetitive high-speed vertical motion, *e.g.*, for chopping scallions (Figure 5a-b), or to mimic a human's tapping motion on the spice bottle to spread enough white pepper (Figure 5c-d). With a spur-gear mechanism, the Roman gripper enables a high speed of rotation of the whisk to mix an egg (Figure 6d-e).

4.1.2 Expanding range of motion. Roman helps expand the range of motion while manipulating specific objects. For example, grasping a screwdriver at certain distances/angles may prevent a conventional robotic arm from performing the rotating manipulation (Figure 3). Therefore, Roman adopts a bevel gear mechanism that could change the rotational axis of the input, expanding the range of motion of the screwdriver's manipulation (Figure 6a-c). Similarly, by attaching the whisk at a small angle (as opposed to being perpendicular) when fastened to the output spur gear, we can expand the range of rotational motion at its end (Figure 6de).
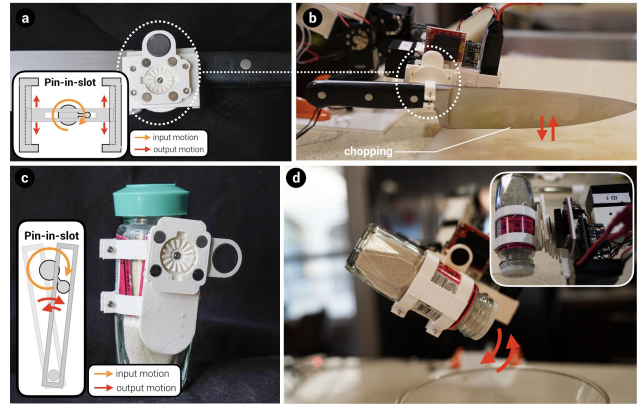

Figure 5: Configuration of the pin-in-slot mechanism (a) to produce periodic up and down motion with a knife (b) and an alternative configuration of the pin-in-slot mechanism (c) to produce periodic side to side motion for spreading peppers from a spice bottle (d).
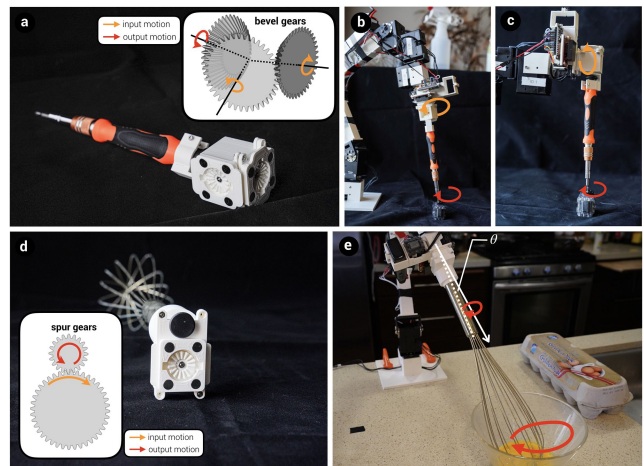

Figure 6: Configuration of the bevel gear mechanism to produce rotational motion at an angle on a screwdriver (a), allowing for two attachment configurations (b, c); the spur gear mechanism with a reverse reduction gear ratio produced higher rotation speeds for a whisk (d) and tilting it by a small angle further expands the range of motion at its end (e).

4.2 Manipulating Objects by Constituent Parts

4.2.1 Enabling squeeze & release manipulation. As shown in Figure 1, the wire cutter augmented with a gear-rack mechanism can perform the task of cutting a wire by squeezing the handles (Figure 1h). Similarly, a gear-rack is used for the spray bottle and thus a robotic arm could fetch the bottle and water the flower automatically (Figure 7). As the spray bottle only requires one-directional manipulation, the mechanism only needs to squeeze in one direction before releasing the handle to return to its original position.

The chopsticks require a bi-directional squeezing and releasing manipulation to pick and place food and a spur gear mechanism
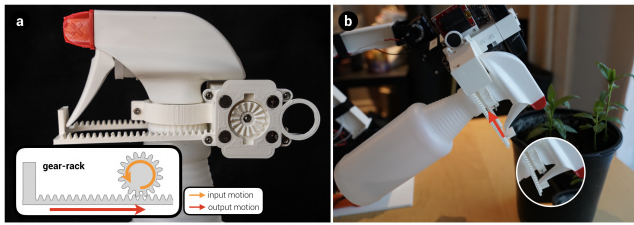
Figure 7: Configuration of the gear rack mechanism to produce linear motion (a) in a single direction to squeeze the handle of a spray bottle (b).
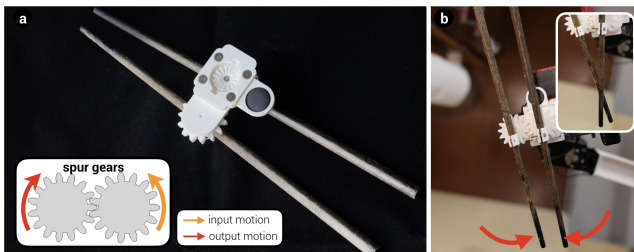

Figure 8: Configuration of the spur gear mechanism to produce rotational motion to squeeze together the tips of chopsticks (a), and a practical demonstration of the chopsticks being manipulated with the mechanism attached (b).
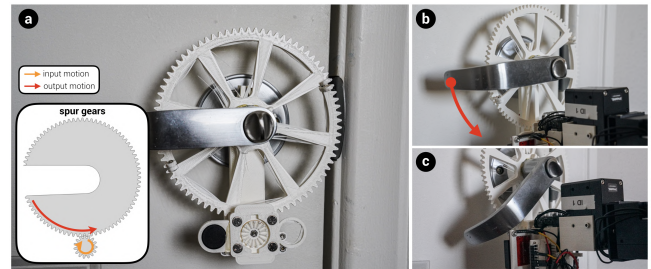

Figure 9: Configuration of the spur gear mechanism with a reduction gear ratio to produce rotational motion with high torque in order to rotate the door knob (a), and a practical demonstration of the door knob being twisted with the mechanism attached (b, c).

Figure 10: Two examples of the bevel gear mechanism to produce rotational motion at an angle to unscrew the lid (a, b) or to rotate the pepper grinder (c, d).

Figure 11: Configuration of the gear rack mechanism to produce bi-directional linear motion in order to squeeze a bottle of hand sanitizer (a), and a practical demonstration of the bottle of hand sanitizer being squeezed with the mechanism attached (b, c).

is used for the manipulation (Figure 8). A gear-rack mechanism is utilized for the can opener to perform the squeezing and rotating manipulation to pierce the peanut butter can (Figure 12d-g). Specifically, a ratchet design (Figure 12e-f) is used to 'lock' the squeezing of the handles (Figure 12h), which then enables the gripper to detach from the handles, attach to the cutting part (Figure 12c), and drive the rotation of the handle to open the can (Figure 12h).

4.2.2   Enabling twist manipulation. Following the manipulation of piercing the can, the robotic arm is able to twist the handle continuously to open the can by using a spur gear mechanism (Figure 12ch). Since both the squeezing and the twisting manipulation of the can opener require a relatively large strength to manipulate, a gear box with ratio of 9:1 is adopted to increase the output torque applied to the target object (Figure 19d). Using a pepper grinder is important in a series of cooking tasks (*e.g.*, making an omelette). With a bevel gear mechanism, the robotic arm can twist the grinder repetitively to sprinkle pepper on the eggs (Figure 10c-d). A robotic arm can also collaborate with a human in a cooking task by opening the lid of a starch jar with a bevel gear mechanism (Figure 10a-b). Roman also enables a robotic arm to open the door by twisting the door knob with a spur gear mechanism on it (Figure 9).

4.2.3   Enabling pump manipulation. With a gear-rack mechanism, an oil spray can be manipulated by a robotic arm to help human cook (Figure 14). A robotic arm can fetch a bottle of hand sanitizer augmented with a gear-rack mechanism and pump it when the user approaches it (Figure 11). To enable a repetitive manipulation of the balloon pump, a gear-rack mechanism is also adopted to perform a bi-directional manipulation (Figure 13).

Figure 13: Configuration of the gear rack mechanism to produce bi-directional linear motion in order to actuate a pump (a), and a practical demonstration of the pump being actuated with the mechanism attached (b, c).

Figure 12: Combination of two separate mechanisms with a reduction gear ratio to produce high torque: a spur gear to cut around the can (c), and a gear rack to pierce the can (also using spur gears to increase torque) (d). A ratchet mechanism (f) is used to maintain the position of the piercing mechanism (*i.e.*, keep the handles squeezed), which allows the gripper to switch to the other mechanism (c) for cutting the can open.

Figure 14: Configuration of the gear rack mechanism to produce linear motion in order to depress the spray button (a), and a practical demonstration of the button being squeezed with the mechanism attached (b, c).
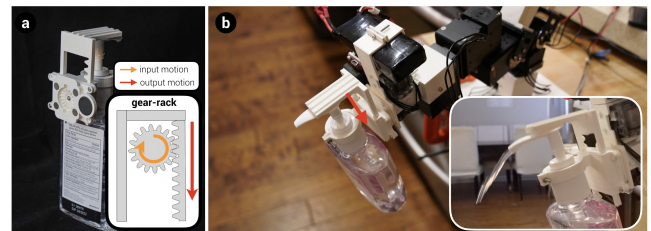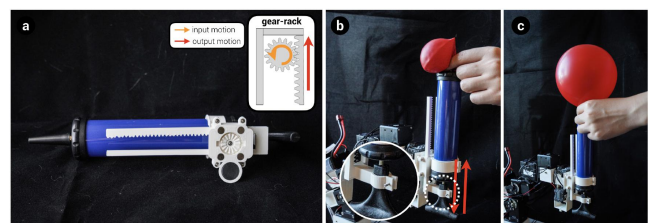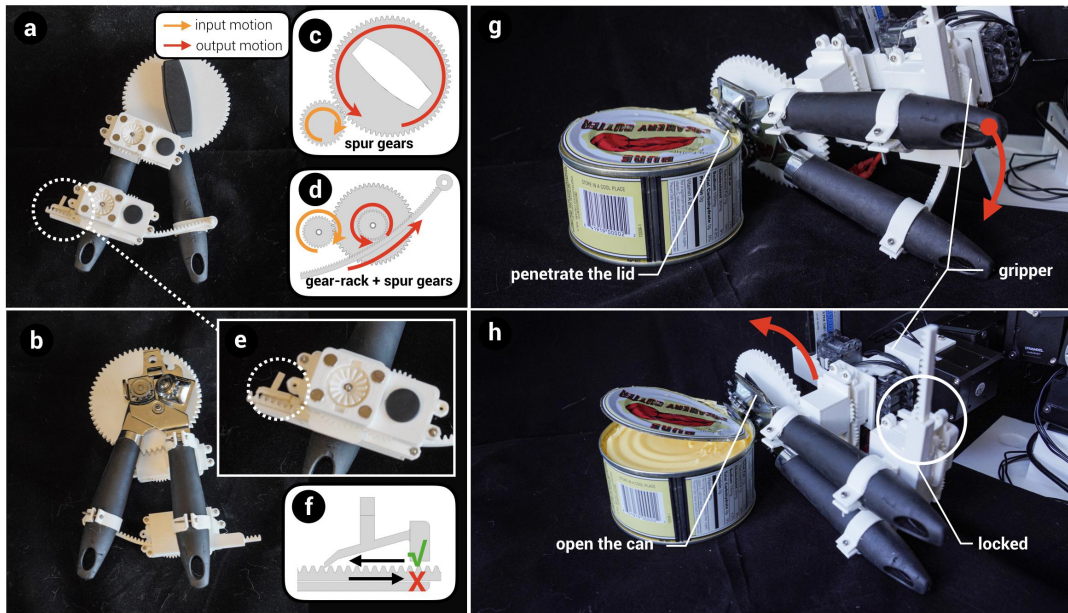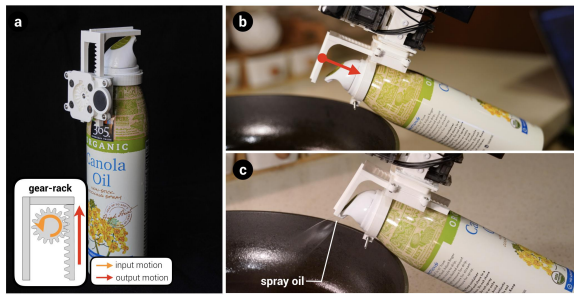
5 SYSTEM OVERVIEW OF ROMAN

Roman is an all-in-one solution to make everyday objects manipulable by generic robotic arms and includes both hardware and software support:

- **Hardware modules** consist of (as shown in Figure 15)
  - **§6.1**: A modular magnetic gripper that can attach to or detach from an object's add-on mechanism, recognize the object to retrieve the corresponding control program, and transfer the driving force from the robotic arm's motor to the mechanism to execute the object-specific manipulation;

- **§6.2**: 3D-printable powerless mechanisms (spur gear, bevel gear, gear-rack, and pin-in-slot) attached to the object which enables objects to be manipulated as a whole or by their constituent parts. The mechanisms are easy to remove/assemble using screws.
- **§7**: **Software module** is a tool for robotic engineers to interactively specify custom motion profiles for manipulating a specific object (*e.g.*, amplitudes of a signal over time to be sent to the motor that drives the mechanisms to squeeze a wire cutter). We specifically focus on authoring motion profiles, which, to the best of our knowledge, is unsupported by prior work; meanwhile, the task of generating the 3D models of mechanisms—based on the type of motion and an object' geometry—can be supported by existing tools [6, 8, 32, 33].

6 HARDWARE IMPLEMENTATION

6.1 A Magnetic Gripper to Attach to, Recognize, and Transfer Motion to an Object's Add-on Mechanism

The magnetic gripper serves as the intermediary between the robotic arm with two main functionalities: 1) attaching to and detaching from the passively actuable mechanism on the target objects and 2) driving the mechanisms on the target objects to perform the manipulation. Further, the gripper also contains an RFID reader for recognizing which object it is attached to and running the corresponding control program.

6.1.1 Attaching and detaching mechanisms: The gripper uses four neodymium magnets (Figure 16D) to generate the magnetic force

Figure 15: Overall hardware structure of Roman
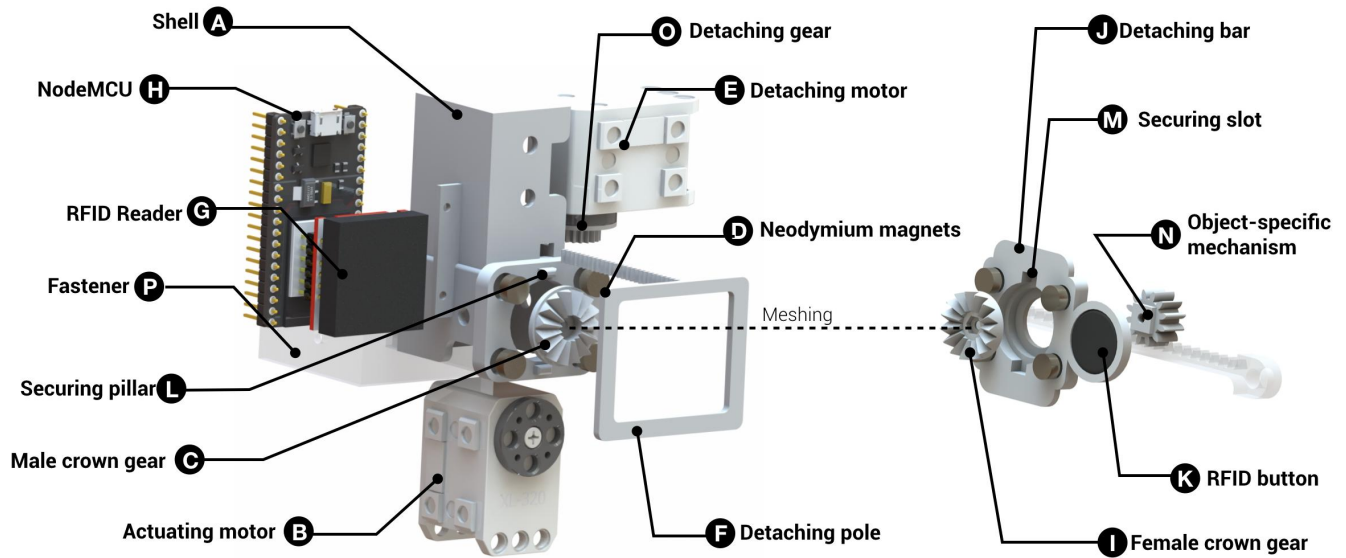

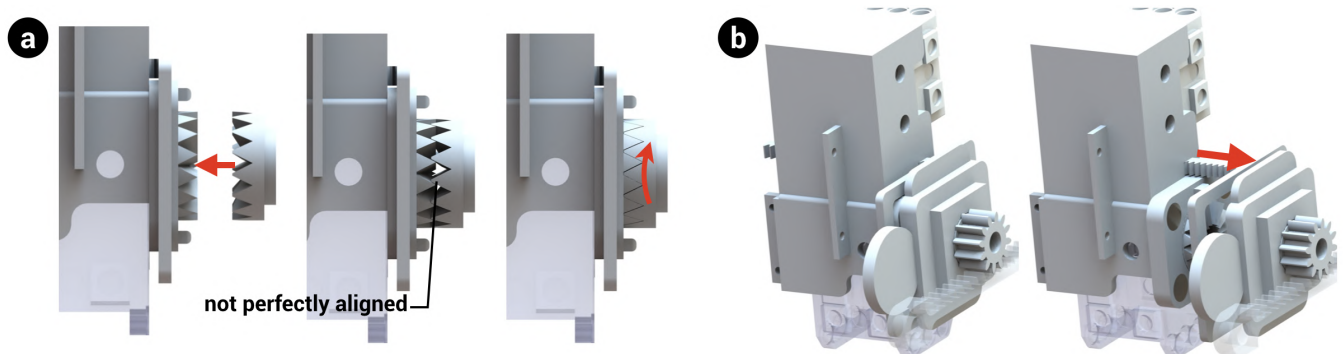**Figure 16: Exploded view of the Roman gripper.**

Figure 17: Meshing (a) and detaching (b) operations of Roman gripper.

for attaching to the mechanism on the target objects. The four neodymium magnets could generate a pull force (the vertical force required to pull the magnets from a steel surface) of 11.2lbs in total, which is sufficient to securely attach to common everyday

handheld objects. Furthermore, the strong magnetic force enables the gripper to attach to objects when it is within approximately 1cm of them, which increases the fault tolerance of the robotic arm's manipulation (*e.g.*, a low cost robotic arm[3] may have accuracy larger than 5mm). To further strengthen the connection, we designed a pin structure (Figure 16L) to counter the lateral force generated during the actuation of the mechanisms.

On the other hand, a stronger connection means larger force required for the detachment. To provide auto-detachment of the mechanisms, Roman employs a Dynamixel XL-320 motor[4] (Figure 16E) and a gear-rack mechanism (Figure 16F/O), which could transfer the rotational motion of the motor into linear motion, for the auto-detachment (Figure 17b). While the XL-320 motor can generate a maximal torque of 0.39 N·M, we designed the gear of the gear-rack mechanism to have a radius of 5mm which enables it to generate 17.52lbs of force on the rack for detachment (Figure 16O).

6.1.2 Motion transmission: The gripper also serves as the actuator of the mechanisms on target objects. Same as the detachment, Roman employs an XL-320 motor (Figure 16B) and a pair of crown gears (Figure 16C/I). The crown gear could transmit the rotational motion of the motor to the mechanism with auto-alignment of the teeth (Figure 17a). On the side of the mechanism, modular driving gears (*e.g.*, gears or gear-rack mechanism, Figure 16N) can be assembled with the female crown gear to actuate different types of mechanisms.

6.1.3 Communicating with target object. Roman employs NodeMCU ESP-12 module[5] for communicating with the web server (Figure 16H). An RFID reader is used for recognizing different objects (Figure 16G) and each mechanism on the object comes with an RFID tag (Figure 16K) whose ID is associated with a user-defined control program to perform a specific task. Fasteners can also be customized and 3D printed to fit different robotic arms using screws (Figure 16P). Finally, the gripper can be powered by an additional 7.4v LiPo battery. The whole system weighs 110g without the battery, which makes it possible to be installed and used on any generic robotic arm.

6.2 Mechanisms for Manipulating Handheld Objects with Different Motion Profiles

We first discuss the mechanism design on the objects' side that transfers the rotary input from the motor into task-specific motion profile for different objects.

Objects require different motion profiles in order to perform their tasks. The motion profile is defined as the required output motion for the objects in order to perform an object-specific task. For example, objects in the *squeezing* category may require a curved motion profile (*e.g.*, the legs of the wire cutter move in an arc trajectory for a cutting task Figure 1); objects in the *twisting* category require a rotary motion profile (*e.g.*, rotating the door knob to open the door Figure 9) and objects in the *pumping* category require a linear motion profile (*e.g.*, linear pushing motion for the balloon pump Fig Figure 13).

---

[3]PincherX 150 Robotic Arm: https://www.trossenrobotics.com/pincherx-150-robot-arm.aspx

[4]https://www.robotis.us/dynamixel-xl-320/

[5]https://www.nodemcu.com/

To address this, Roman adopts four basic types of mechanical mechanisms: spur gears, bevel gears, gear-and-rack and pin-in-slot mechanisms (Figure 18). As the motor outputs rotary motion, the goal of the selected mechanisms is to transfer the rotary motion into desired motion profiles. Comparing to the mechanisms used in [33], Roman only focuses on the motion output while selecting the mechanisms and uses additional gearbox to tackle the torque requirement.

6.2.1 Spur gears. Spur gears are a mechanism in which multiple gears mesh together to transmit the rotary motion from one shaft to another (Figure 18a). Therefore this mechanism can transfer the rotary motion from the motor input into a rotary motion profile.

Rotary to rotary While the output motion has the same rotary motion as the input, the gear-pair mechanism can translate the rotary axis to a parallel position, which enables the mechanism to be anchored to a fixed point on the object while manipulating the object. As shown in Figure 9, instead of being directly anchored to the door knob, which may get in the way of people using it, the mechanism for driving the knob can be shifted to a position where it does not interfere with regular use of the knob.

Speed and strength Besides the translation of the rotational axis, Roman leverages the property of reduction in gears to generate higher speed or strength than is typically available from a motor. Gear reduction is an arrangement of gears in which an input speed can be raised at the expense of torque, or the output torque can be raised at the expense of speed as explained in the following equations:

$$\text{Speed}_{\text{output}} = \frac{r_{\text{input}}}{r_{\text{output}}} \times \text{Speed}_{\text{input}}$$

$$\text{Torque}_{\text{output}} = \frac{r_{\text{output}}}{r_{\text{input}}} \times \text{Torque}_{\text{input}}$$

With a proper selection of the gears, the mechanism can generate much larger torque than the motor's base ability (Figure 19). In the meantime, with a reversed reduction gear, the mechanism can generate higher rotational speed than the motor's typical maximum speed. The whisk is an example of leveraging reversed reduction gears to enable the whisk to rotate at a high speed to beat the egg (Figure 6de). Such reduction gears can also be combined with other mechanisms depending on the task (*e.g.*, the can-opener adopts a reduction gear box over its gear-rack mechanism to generate enough force to pierce the can as shown in Figure 12d).

6.2.2 Bevel gears. Similar to the spur gears, the output of the bevel gears is also rotary motion. Unlike spur gears, bevel gears change the orientation of the rotational axis, which enables robotic arms to execute a twisting task from different angles (Figure 18b).

Rotary to rotary Bevel gears can generate a rotary motion profile for target objects. With the property of being able to change the orientation of the rotational axis, bevel gears enable robotic arms to operate twisting tasks with a space-efficient solution. For example, in order to perform a lid opening task, a spur gears design would make the mechanisms bulky as the driving gear will protrude from the jar lid. In contrast, A bevel gears design rotates the protruding gear so that it reduces the overall volume occupied by the entire mechanism (Figure 10).

a Spur gear
b Bevel gear
c Gear-rack
d Pin-in-slot
without pivot:
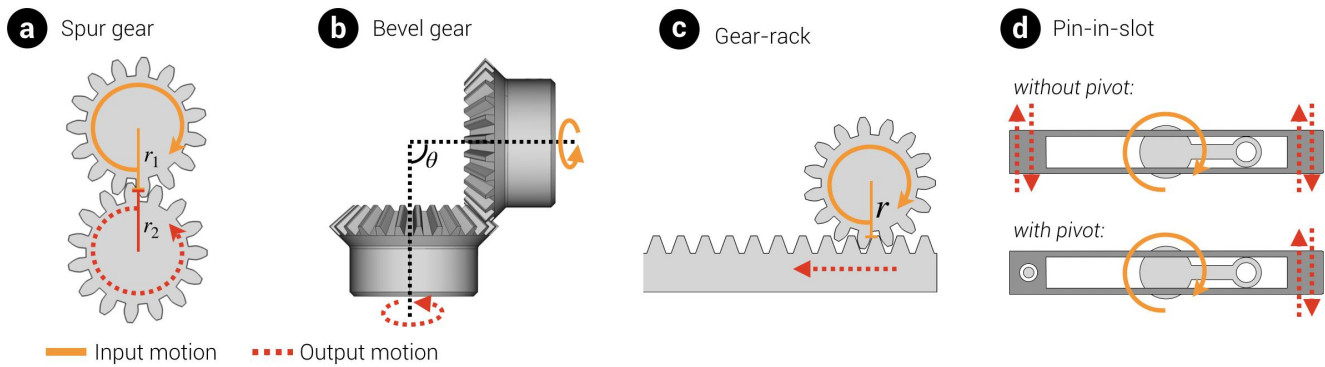*with pivot:*

— Input motion   ···· Output motion

Figure 18: Overview of the four types of mechanisms used to achieve target motions: spur gears for rotational motion (a), bevel gears for rotational motion at an angle (b), gear rack for linear motion (c), and a pin-in-slot mechanism to achieve periodic motion (d).

a speed increased $r_1 > r_2$
b torque increased $r_1 < r_2$
c gearbox design
d input / output

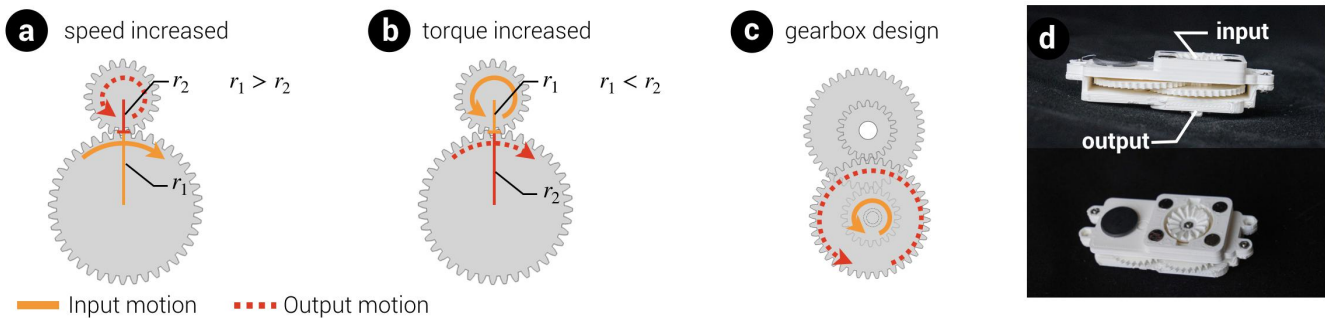— Input motion   ···· Output motion

Figure 19: Breakdown of the configuration of the gearbox, showing how to produce higher speeds at the expense of torque (a) and to produce higher torque at the expense of speed (b), as well as the physical configuration of the gears (c) and the design of the gearbox (d).

6.2.3 Gear-rack. The gear-rack mechanisms are utilized to convert the rotary motion from the gear into the linear motion of the rack (Figure 18c). As the gear-rack mechanism generates a linear motion profile, this mechanism can be used by the objects in the *grasp & pump* category. Further, the gear-rack mechanism can also be utilized for generating a curved motion profile, which will be discussed below.

Rotary to linear While the gear-rack mechanism converts the rotary motion from the gear into the linear motion of the rack, it could help the robotic arm to manipulate objects that require pumping (Figure 4E). Designers of the mechanism could adjust the output velocity and force by modifying the size of the gear based on the equation:

$$\text{Speed}_{\text{output}} = \text{rpm}_{\text{motor}} \times r$$

$$\text{Force}_{\text{output}} = \frac{\text{torque}_{\text{motor}}}{r}$$

where r represents the radius of the gear attached to the motor. For example, hand sanitizer is an example that requires large torque to squeeze out sanitizer Figure 11.

Rotary to curved Besides the linear motion profile, the gear-rack is able to generate a curved motion profile for squeezing objects such as the squeezing leg of can opener (Figure 12). Roman leverages

the compliant property of the PLA material to design racks that can bend themselves to adapt to the curved motion (Figure 12e). As a result, the fastener on the rack (normally attached with the other movable part of the object) would require an active joint for the rack to rotate relatively, without which the curved motion may generate a large offset at the tip of the rack and break the mechanism.

Bi-directional manipulation The gear-rack mechanism may deal with objects requiring either one or bi-directional manipulation. For objects with only one-directional manipulation such as the oil spray (Figure 14), a simple bar is sufficient for generating a one-directional force. However, objects that require bi-directional manipulation such as the balloon pump, require an additional structure fastened to the part to enable motion in two directions (Figure 13b).

Single-direction constraints Roman also provides a single direction constraint using the gear-rack mechanism. For example, in order to perform the task of piercing the can and then opening it, the mechanism needs to lock the squeezing mechanism for the opener to continue to pierce through the surface of the can. To achieve this, we employ a design inspired by the ratchet gear, where the rack can move freely in one direction while being prevented from moving in the opposite direction unless a pin is depressed, which releases the mechanism (Figure 12d).

6.2.4 Pin-in-slot. A pin-in-slot is a mechanism where a pin-joint moves along or parallel to a slide-joint (Figure 18d). The pin-joint receive the rotary input from the motor and transfer the motion to the slide-joint. By fixing one side of the slide-joint, the pin-in-slot mechanism can generate a single-sided motion profile (Figure 18d bottom) while a double-sided motion profile can also be generated by placing the slide-joint inside of another slider joint (Figure 18d top).

The spice bottle is an example of using the pin-in-slot mechanism to mimic the human shaking action by generating a single-sided motion profile (Figure 5cd) and the knife is an example of performing the chopping task by generating a double-sided motion profile (Figure 5ab).

Periodic motion Besides generating linear and curved motion profiles, the pin-in-slot mechanism is good at generating a periodic motion. As shown in Figure 20, the continuous rotation of the pin-joint can generate a double-sided or single-sided periodic motion at the slide-joint, *e.g.*, for continuously shaking a spice bottle (Figure 5cd). While it is possible for other mechanisms to generate a periodic motion with periodically changed control input, the control signal may experience data loss when changing intensely in short periods. Differently, the periodic motion generated by the pin-in-slot mechanism is easier to control because it relies on the stability of the mechanism components.

7 SOFTWARE IMPLEMENTATION: A TOOL FOR ROBOTIC ENGINEERS TO SPECIFY CUSTOM MOTION PROFILES FOR OBJECT-SPECIFIC MECHANISMS

In this section, we illustrate the workflow of a user authoring the control program for a task given some pre-generated mechanisms to manipulate a target object. We assume that the users are robotic engineers with a mechanical or robotic background and the mechanism generation is supported elsewhere by tools similar to [8, 21, 33]. We exclusively focus on the less-supported part where the users author control programs for different object-specific tasks. Roman's front end is written in JavaScript and the back end is written in Python. Roman communicates with the ESP8266 module on the NodeMCU and stores the motion profile through a Python-based web server. The RFID reader on the gripper reads the ID of different RFID chips on the objects and accesses the corresponding motion profile via the NodeMCU.

7.1 Custom Motion Profiles

As Roman transfers the rotary motion input from the motor to a customized output motion profile, it requires further specification of the motion profiles in order to perform an object-specific task. For example, to design a task for manipulating the wire cutter while collaborating with a human user (Figure 1), the wire cutter is expected to *(i)* move to the desired position for the user to hold the wire for cutting, *(ii)* squeeze the handles about half-way and hold the position in order for the user to align and adjust the wire, *(iii)* squeeze fully to cut the wire, and *(iv)* return the handles to the initial configuration. While step *(i)* is conducted by the movement of the robotic arm, the rest are done by Roman hardware and require a custom design of the motion profile. More specifically, the

customizability of the motion profiles amounts to specifying the amplitudes u of the control signal over time, which corresponds to the rotational speed of the motor output. The aforementioned task for the wire cutter has a motion profile as the control signal shown in Figure 22.

7.2 User Interface

As shown in Figure 21, to facilitate the design of such custom motion profiles, the Roman software provides a user interface for interactively specifying the custom motion profiles by adding or adjusting the key points in the graph of a control signal (Figure 21c). The u for the y-axis is a unitless value ranged from -1 to 1 which represents the rotational speed of the motor relative to the maximum speed under the current load (with negative values corresponding to rotation in the opposite direction). The x-axis represents time and the user has the ability to modify the total time the motor is running at a certain speed by adjusting the length of each period of the control signal. With the combination of different control signals, the user could design a custom motion profile for a object-specific task.

7.2.1 Motion Profile Templates and Interactive Graph Editing. Roman provides different ways to interact with the graph of the control signal. As shown in Figure 21a and d, Roman provides four types of motion profile templates for a user to adapt to their needs, including endless rotation (*e.g.*, whisk), periodic motion (*e.g.*, knife), one-way motion (*e.g.*, jar lid), and two-way motion (*e.g.*, wire cutter). By clicking on the corresponding template, the control signal is imported into and visualized on the interactive graph and the user can further adjust the motion profiles.

To start editing the graph, a user simply double-clicks anywhere on the graph and then adjusts the position of an existing point by dragging it. The user can also adjust the range of the x-axis (*i.e.*, increase/decrease time) by clicking on the +/- signs (Figure 21b), which either adds or removes 1 second.

7.2.2 Real Time Testing. There is uncertainty in manipulating real-world objects, *e.g.*, the manipulation of a wire cutter (the motor's rotational speed) might need to vary with different wires. As such, Roman enables a real-time *testing* mode (Figure 21b) that creates a feedback loop in which the user could test and adjust the custom motion profile to see how well the manipulation is performed on the target object. The user could also check the box of 'continuous' to specify a repetitive motion. Once a user has tested and is satisfied with the motion profile, they can click on the 'save' button to store the motion profile to the target object.

8 VALIDATION WITH EXPERT INTERVIEW

We conducted an expert interview as an initial step to evaluate Roman. The purpose of this interview is to evaluate the usefulness and practicality of Roman in helping robotic engineers achieve object manipulation tasks, as well as to gather feedback and further suggestions from experts in the field of robotics.

8.1 Participants & Procedure

Four participants (all male with an average age of 28.25) with expert knowledge in the domain of robotics and mechanism design

Figure 20: Demonstration of the two types of periodic motion that could be accomplished with a pin-in-slot mechanism: angular (a), and up and down (b).

Figure 21: The Roman user interface used for selecting motion templates, authoring control programs, and uploading them to the robotic arm.

were invited for interviews. Three of them are Ph.D. students in the Mechanical Engineering Department with a focus on the research of robotic design and control (P1-P3). Among these, two had prior experience in developing robotic manipulators as a product in a robotic start-up (P1 and P2). The other participant is a post-doc researcher in Electrical Engineering Department with a Ph.D. degree in mechanical engineering and his research focuses on the computational design for origami robots (P4). Although no participants

Figure 22: A sample motion profile used to control the wire cutter in order to cut a wire.

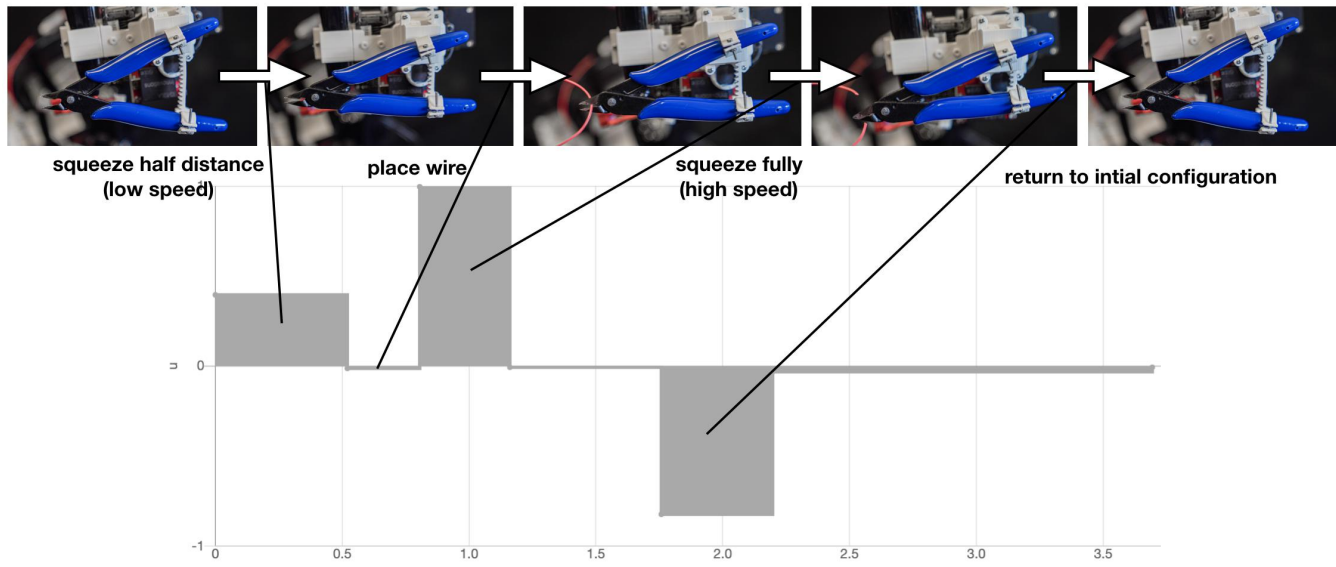worked in the exact area of robotic manipulation in a human environment, all were familiar with the concept of mechanism design and control of robotic arm manipulation.

Each of the participants was interviewed in-person. First, they were explained the goal and function of Roman hardware and one author showed them demos of using the Roman gripper to manipulate five different objects (wire cutter, hand sanitizer, jar lid, whisk and spice can). These objects covered the four types of mechanisms we used in Roman as well as the four motion profile templates in the user interface. Afterwards, they were introduced to the Roman software and were asked to replicate the custom motion profiles of the wire cutter (which has the most complicated motion profile as shown in Figure 22). The motion profiles designed by the participants were test by performing a cutting task (similar to Figure 1) of an AWG 18 wire. The participants were able to design the profile and cut the provided wire physically in only 5 minutes (excluding the time needed to become familiar with interacting with the user interface) We then conducted a semi-structured interview, mainly to collect their feedback on the capability and practicality of Roman and potential directions for future research. The entire study lasted on average 25 minutes per participant.

8.2 Results and Feedback

All the participants were positive to Roman's features, leaving feedback for its *utility* and *practicality* of Roman.

8.2.1 Roman is complementary to current approaches in robotic manipulation in human environment. All the participants considered Roman to be a complement to the current research in the robotic manipulation in situations that require robotic arms in human environment. P2 stated that most of the existing robotic arms have limited functions or they required complicated programming of the control to use different types of tools. He thought that Roman provided a simple solution as the object + mechanism can be

pre-designed as a product and delivered to customers. From the perspective of a user, P2 was initially concerned about the practically of possessing a robotic arm in daily life, which was later answered by himself when he noted that the Roman solution could make a robotic arm adapt to different objects in a task-heavy scenario like cooking. P1 was concerned about the pre-design of the custom motion profiles to be time-consuming, but he also pointed out that in the application scenarios such as cooking, the number of target objects is limited and therefore it is acceptable. P3 considered Roman usable, but he was more concerned about the necessity of the usage of robotic arm in such scenarios versus designing more easy-to-manipulate objects for human (*e.g.*, designing a jar that can be opened by pressing a button instead of adding mechanism to be manipulable by robotic arm). P4 mentioned that it makes more sense to have a strong and functional robotic hand to manipulate different objects without altering them but he agreed that the artificial hand solution is costly (a powerful enough robotic hand normally costs more than $10k) and Roman could be a low-cost solution.

8.2.2 Roman's mechanism designs and custom motion profiles are replicable by robotic experts. When asked about whether they are able to replicate the hardware design of Roman's five examples[6], all the participants agreed and were able to understand the rationale behind the design of each mechanism. P1 thought that it would be easy for a person with fundamental knowledge of mechanical design to replicate all of the examples. P2 also pointed out that with the explanation of different mechanisms as in the previous section (§6.1.3) as a guideline, it is a lot easier to select and design the mechanisms for specific objects. P4 stated that it is easy for robotic experts to replicate the examples but it would be great if a parametric design tool is provided for novice users.

---

[6]Note that participants did not actually replicate those examples, considering the time-consuming process of modeling, printing and assembly.

All the participants were able to replicate the custom motion profiles for the wire cutter to perform a wire cutting task. All the participants could understand the differences between each motion profile template. P2 also valued the provided templates because, with the templates, he could know the overall movement of the mechanism and would only need to adjust a few parameters.

8.2.3 Roman's hardware could be further improved. The participants suggested that the hardware components can be improved to make the overall structure smaller and more durable including developing new types of mechanism (making the mechanisms more versatile), replacing the neodymium magnets with electromagnets (making the mechanisms smaller) and printing the custom mechanism in carbon (making the mechanisms stronger).

In summary, results of the expert interview indicated that Roman could be a potential solution for robotic manipulation to assist human with daily tasks in certain scenarios. Also, it was considered practical for people with a robotic or mechanical engineering background to design and replicate the objects with mechanisms using Roman. The participants also recommended that a tool for novice users to custom such mechanism for existing objects would be valuable.

9 LIMITATIONS, TRADEOFFS, FUTURE WORK

9.1 Performance testing

Roman's main contribution is the idea of adding mechanisms to enhance objects' manipulability, which complements existing approaches focused on gripper design and manipulation algorithms. As such, we chose to validate Roman by creating various examples to show the idea's wide applicability and by interviewing four robotic experts to obtain their initial feedbacks and reactions that would set the scene for future work. As next steps, future work that uses Roman on a specific type of objects/tasks (*e.g.*, manipulating a wire cutter) should define metrics for success of each task (*e.g.*, a wire is completely cut in one trial), control pertinent variables (*e.g.*, the thickness of wires), and employ repeated measures to obtain such metrics.

9.2 Mechanisms interfering with normal use

As Roman attaches 3D printed mechanism onto the existing object, some of the mechanisms may interfere with the normal use of the objects. For example, the spur gear mechanism on the chopsticks obstruct the way of a human would use them (Figure 8) and the bulky mechanism on the can opener also makes it hard for human to manipulate (Figure 12). While some of the mechanism can be easily disassembled, *e.g.*, the rack of the hand sanitizer can be easily pulled out (Figure 11), future work can focus on making the mechanisms modular and easy to disassemble such as LEGO MindStorm [27]. It is also possible to embed such function during the product design stage that aims at making the overall object + mechanism manipulable by both humans and robotic arms.

9.3 Incorporating sensing & perception

Roman focuses on enhancing manipulability by enabling human collaborating with Roman-equipped robotic arm or a Roman-equipped robotic arm to perform tasks independently. However, the latter scenario assumes that the object's position and orientation are known, which is a trade-off of not integrating sensing modules in the current design. Given the plethora of work on sensing and perception (*e.g.*, [49]), future work can add such modules to Roman's hardware components, which are expected to work independently and complementarily to the current set-up.

9.4 Generalizability of the design

Currently, Roman only focuses on tasks driven by the motor of the gripper while the robotic arm have to be manually configured. As such, a Roman mechanism only affords manipulation in a limited space and cannot enable large-scale tasks, *e.g.*, holding a spatular to make stir-fry.

Future work may extend Roman to include software support involving the entire robotic arm, *e.g.*, linking the action of the robotic arm with the actuation of the mechanism. With that, the Roman hardware could collaborate with the robotic arm to achieve more complicated tasks *e.g.*, scooping ice cream (enabled by Roman mechanisms) and distributing it into different locations (enabled by the rest of the robotic arm) automatically.

On the control side, Roman uses open-loop control of the mechanisms. This limits Roman to manipulate objects that have dynamic feedback such as tightening screws in different conditions. One of the future directions is adding haptic sensors to the mechanisms and incorporate feedback control while performing the tasks.

9.5 Trade-off between torque and the size/complexity of mechanisms

Currently, Roman enables the manipulation of objects by generating motions using a low-cost motor and 3D printed mechanisms. As a result, Roman is limited to supporting objects that do not require a lot of torque. While Roman provides a partial solution by adding a gearbox in between the gripper and the object to increase the torque (see the can opener example Figure 12), there is a trade-off between the size of the mechanism and the maximum torque the gripper could generate. For example, manipulating a hedge trimmer (too large) or a sealed jar (too tight) will require rather bulky mechanisms unrealistic to be attached to the object. Substituting the motors in the current design with stronger models could partially solve the problem. Alternatively, there exist research opportunities to solve this problem with improved mechanisms, *e.g.*, using more durable materials such as metal linkages or cables to increase the generated force.

9.6 Manipulating objects with multiple movable parts or multiple consecutive manipulations

Some objects are articulated with multiple movable parts, *e.g.*, multifunctional pliers, Swiss army knife, flexible selfie rods, and Rubik's cubes. Technically, Roman's mechanisms can extend to more parts by enabling one additional motion at a time, yet, practically, too many mechanisms might not be fittingly added to an object and might even interfere with one another. Future work could focus on involving different materials (cable, metal, carbon) to make the mechanism smaller while functional.

Other objects might require consecutive manipulations to perform a task, *e.g.*, a corkscrew requires a twisting motion on the handle to penetrate a wine cork and a subsequent squeezing motion on the two arms, followed by pulling out the same handle. Currently Roman needs to provide separate mechanisms for each motion, which could result in the overall mechanisms too bulky. To address this problem, future work could incorporate interactive trajectory design into the mechanism design of Roman, similar to the approach in [11] that would need only one custom mechanism to accomplish multiple consecutive manipulations.

9.7 Opportunities to support novice users

As mentioned by the robotic engineers in the interview, while it is easy for people with knowledge of robotics to replicate examples in Roman or design mechanism for new examples, novice users might struggle to design mechanism and understand which mechanism to use and how to specify the control signals since Roman does not provide 3D modelling support. One possible idea for future work is to instrument sensors on an object similar to the current Roman mechanisms. Then a user can demonstrate how they would manipulate an object, which can be captured by such sensors (*e.g.*, inertial measurement units or reflective markers + external optical tracking) and incorporate the algorithm in Robiot [33] which could generate the mechanism and control program automatically. In this way, Roman would be more usable to novice users.

9.8 Areas of hardware improvement

The hardware of Roman can be improved by: *(i)* switching to a half-duplex enabled microcontroller board that could enable the control of the mechanism to be self-adaptive to different loading conditions; *(ii)* replacing the neodymium magnets with electromagnets that can be programmatically controlled for attachment/detachment, thus dispensing with the additional motor currently used for detachment.

ACKNOWLEDGMENTS

REFERENCES

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 3–20.

[2] Vincent Babin and Clément Gosselin. 2018. Picking, grasping, or scooping small objects lying on flat surfaces: A design approach. *The International Journal of Robotics Research* 37, 12 (2018), 1484–1499.

[3] Vincent Babin and Clément Gosselin. 2020. Mechanisms for Robotic Grasping and Manipulation. *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2020).

[4] Vincent Babin, David St-Onge, and Clément Gosselin. 2019. Stable and repeatable grasping of flat objects on hard surfaces using passive and epicyclic mechanisms. *Robotics and Computer-Integrated Manufacturing* 55 (2019), 1–10.

[5] Ron Berenstein, Averell Wallach, Pelagie Elimbi Moudio, Peter Cuellar, and Ken Goldberg. 2018. An open-access passive modular tool changing system for mobile manipulation robots. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 592–598.

[6] Xiang'Anthony' Chen. 2016. Making Fabrication Real. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 17–20.

[7] Xiang'Anthony' Chen, Stelian Coros, and Scott E Hudson. 2018. Medley: A library of embeddables to explore rich material properties for 3D printed objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[8] Xiang'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E Hudson. 2016. Reprise: A design tool for specifying, generating, and customizing 3D printable adaptations on everyday objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 29–39.

[9] Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 29–39. https://doi.org/10.1145/2984511.2984512

[10] Andrea S Ciullo, Janne M Veerbeek, Eveline Temperli, Andreas R Luft, Frederik J Tonis, Claudia JW Haarman, Arash Ajoudani, Manuel G Catalano, Jeremia PO Held, and Antonio Bicchi. 2020. A novel soft robotic supernumerary hand for severely affected stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28, 5 (2020), 1168–1177.

[11] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

[12] Scott Davidoff, Nicolas Villar, Alex S Taylor, and Shahram Izadi. 2011. Mechanical hijacking: how robots can accelerate UbiComp deployments. In *Proceedings of the 13th international conference on Ubiquitous computing*. 267–270.

[13] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. 2017. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3530–3537.

[14] Alberto Elfes. 1989. Using occupancy grids for mobile robot perception and navigation. *Computer* 22, 6 (1989), 46–57.

[15] Erik D Engeberg, Nancy Pollard, Maximo Roa, and Zeyang Xia. 2018. Robotic grasping and manipulation competition: task pool. *Robotic Grasping and Manipulation: First Robotic Grasping and Manipulation Challenge, RGMC 2016, Held in Conjunction with IROS 2016, Daejeon, South Korea, October 10–12, 2016, Revised Papers* 816 (2018), 1.

[16] T. Feix, J. Romero, H. B. Schmiedmayer, A. M. Dollar, and D. Kragic. 2016. The GRASP Taxonomy of Human Grasp Types. *IEEE Transactions on Human-Machine Systems* 46, 1 (2016), 66–77. https://doi.org/10.1109/THMS.2015.2470657

[17] David Fischinger, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann, Tobias Koertner, Astrid Weiss, Antonis Argyros, et al. 2016. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems* 75 (2016), 60–78.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

[19] Yoav Golan, Amir Shapiro, and Elon Rimon. 2020. Jamming-free immobilizing grasps using dual-friction robotic fingertips. *IEEE Robotics and Automation Letters* 5, 2 (2020), 2889–2896.

[20] David Gyimothy and Andras Toth. 2011. Experimental evaluation of a novel automatic service robot tool changer. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 1046–1051.

[21] Sehoon Ha, Stelian Coros, Alexander Alspach, James M Bern, Joohyung Kim, and Katsu Yamane. 2018. Computational design of robotic devices from high-level motion specifications. *IEEE Transactions on Robotics* 34, 5 (2018), 1240–1251.

[22] Alaa Hassan and Mouhammad Abomoharam. 2014. Design of a single DOF gripper based on four-bar and slider-crank mechanism for educational purposes. *Procedia CIRP* 21 (2014), 379–384.

[23] Zhengtao Hu, Weiwei Wan, and Kensuke Harada. 2019. Designing a mechanical tool for robots with two-finger parallel grippers. *IEEE Robotics and Automation Letters* 4, 3 (2019), 2981–2988.

[24] Shohei Katakura, Yuto Kuroki, and Keita Watanabe. 2019. A 3D printer head as a robotic manipulator. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 535–548.

[25] Shohei Katakura and Keita Watanabe. 2018. Printmotion: Actuating printed objects using actuators equipped in a 3D printer. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. 137–139.

[26] Charles C Kemp, Aaron Edsinger, and Eduardo Torres-Jara. 2007. Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 20–29.

[27] Frank Klassner and Scott D Anderson. 2003. Lego MindStorms: Not just for K-12 anymore. *IEEE robotics & automation magazine* 10, 2 (2003), 12–18.

[28] Hikmet Kocabas. 2009. Gripper design with spherical parallelogram mechanism. *Journal of Mechanical Design* 131, 7 (2009).

[29] Ioannis Kostavelis, Dimitrios Giakoumis, Georgia Peleka, Andreas Kargakos, Evangelos Skartados, Manolis Vasileiadis, and Dimitrios Tzovaras. 2018. RAMCIP robot: a personal robotic assistant; demonstration of a complete framework. In

Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 0–0.

[30] Eric Krotkov, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orlowski. 2017. The darpa robotics challenge finals: Results and perspectives. *Journal of Field Robotics* 34, 2 (2017), 229–240.

[31] Kiju Lee, Yanzhou Wang, and Chuanqi Zheng. 2020. Twister hand: Underactuated robotic gripper inspired by origami twisted tower. *IEEE Transactions on Robotics* 36, 2 (2020), 488–500.

[32] Jiahao Li, Meilin Cui, Jeeeun Kim, and Xiang'Anthony' Chen. 2020. Romeo: A Design Tool for Embedding Transformable Parts in 3D Models to Robotically Augment Default Functionalities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 897–911.

[33] Jiahao Li, Jeeeun Kim, and Xiang'Anthony' Chen. 2019. Robiot: A Design Tool for Actuating Everyday Objects with Automatically Generated 3D Printable Mechanisms. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology.* 673–685.

[34] Nianlong Li, Han-Jong Kim, LuYao Shen, Feng Tian, Teng Han, Xing-Dong Yang, and Tek-Jin Nam. 2020. HapLinkage: Prototyping Haptic Proxies for Virtual Hand Tools Using Linkage Mechanism. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 1261–1274.

[35] Z-C Lia and C-H Menq. 1988. The dexterous workspace of simple manipulators. *IEEE journal on Robotics and Automation* 4, 1 (1988), 99–103.

[36] Chih-Hsing Liu, Ta-Lun Chen, Chen-Hua Chiu, Mao-Cheng Hsu, Yang Chen, Tzu-Yang Pai, Wei-Geng Peng, and Yen-Pin Chiang. 2018. Optimal design of a soft robotic gripper for grasping unknown objects. *Soft robotics* 5, 4 (2018), 452–465.

[37] Hangxin Liu, Chi Zhang, Yixin Zhu, Chenfanfu Jiang, and Song-Chun Zhu. 2019. Mirroring without overimitation: Learning functionally equivalent manipulation actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8025–8033.

[38] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. 2019. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684* (2019).

[39] R. Mutlu, G. Alici, M. in het Panhuis, and G. Spinks. 2015. Effect of flexure hinge type on a 3D printed fully compliant prosthetic finger. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM).* 790–795. https://doi.org/10.1109/AIM.2015.7222634

[40] Ken Nakagaki, Joanne Leong, Jordan L Tappa, João Wilbert, and Hiroshi Ishii. 2020. HERMITS: Dynamically Reconfiguring the Interactivity of Self-Propelled TUIs with Mechanical Shell Add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 882–896.

[41] Ken Nakagaki, Yingda Liu, Chloe Nelson-Arzuaga, and Hiroshi Ishii. 2020. TRANS-DOCK: Expanding the Interactivity of Pin-based Shape Displays by Docking Mechanical Transducers. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction.* 131–142.

[42] Kento Nakayama, Weiwei Wan, and Kensuke Harada. 2019. Designing grasping tools for robotic assembly based on shape analysis of parts. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids).* IEEE, 1–7.

[43] Francesca Negrello, Werner Friedl, Giorgio Grioli, Manolo Garabini, Oliver Brock, Antonio Bicchi, Máximo A Roa, and Manuel G Catalano. 2020. Benchmarking hand and grasp resilience to dynamic loads. *IEEE Robotics and Automation Letters* 5, 2 (2020), 1780–1787.

[44] H. Nishimura, A. Kakogawa, and S. Ma. 2012. Development of an underactuated robot gripper capable of retracting motion. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO).* 2161–2166. https://doi.org/10.1109/ROBIO.2012.6491289

[45] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. 2000. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, 255–262.

[46] Adam Pettinger, Conner Dimoush, and Mitch Pryor. 2019. Passive tool changer development for an elastic and compliant manipulator. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE).* IEEE, 1200–1205.

[47] Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2016. Retrofab: A design tool for retrofitting physical interfaces using actuators, sensors and 3d printing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* 409–419.

[48] Iason Sarantopoulos, Yannis Koveos, and Zoe Doulgeri. 2018. Grasping flat objects by exploiting non-convexity of the object and support surface. In *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 5606–5611.

[49] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. 2008. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research* 27, 2 (2008), 157–173.

[50] Philipp Schoessler, Daniel Windham, Daniel Leithinger, Sean Follmer, and Hiroshi Ishii. 2015. Kinetic blocks: Actuated constructive assembly for interaction and display. In *Proceedings of the 28th Annual ACM Symposium on User Interface*

Software & Technology. 341–349.

[51] Takeshi Takaki and Toru Omata. 2006. 100g-100N finger joint with load-sensitive continuously variable transmission. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 976–981.

[52] Jun Ueda, Masahiro Kondo, and Tsukasa Ogasawara. 2010. The multifingered NAIST hand system for robot in-hand manipulation. *Mechanism and Machine Theory* 45, 2 (2010), 224–238.

[53] Chen Wang, Daniel Freer, Jindong Liu, and Guang-Zhong Yang. 2019. Vision-based Automatic Control of a 5-Fingered Assistive Robotic Manipulator for Activities of Daily Living. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 627–633.

[54] Zeli Wang, Heng Li, and Xintao Yang. 2020. Vision-based robotic system for on-site construction and demolition waste sorting and recycling. *Journal of Building Engineering* 32 (2020), 101769.

[55] Jingren Xu, Weiwei Wan, Keisuke Koyama, Yukiyasu Domae, and Kensuke Harada. 2021. Selecting and designing grippers for an assembly task in a structured approach. *Advanced Robotics* (2021), 1–17.

[56] Simon Zimmermann, Ghazal Hakimifard, Miguel Zamora, Roi Poranne, and Stelian Coros. 2020. A multi-level optimization framework for simultaneous grasping and motion planning. *IEEE Robotics and Automation Letters* 5, 2 (2020), 2966–2972.