ReCompFig: Designing Dynamically Reconfigurable Kinematic Devices Using Compliant Mechanisms and Tensioning Cables

Humphrey Yang
Human-Computer Interaction Institute, Carnegie Mellon University
hanliny@cs.cmu.edu

Tate Johnson
School of Design, Carnegie Mellon University
tatej@andrew.cmu.edu

Ke Zhong
Department of Material Science and Engineering, Carnegie Mellon University
kezhong@andrew.cmu.edu

Dinesh K. Patel
Human-Computer Interaction Institute, Carnegie Mellon University
dineshpa@andrew.cmu.edu

Gina Olson
Department of Mechanical Engineering, Carnegie Mellon University
golson@andrew.cmu.edu

Carmel Majidi
Department of Mechanical Engineering, Carnegie Mellon University
cmajidi@andrew.cmu.edu

Mohammad Islam
Department of Material Science and Engineering, Carnegie Mellon University
mohammad@cmu.edu

Lining Yao
Human-Computer Interaction Institute, Carnegie Mellon University
liningy@cs.cmu.edu

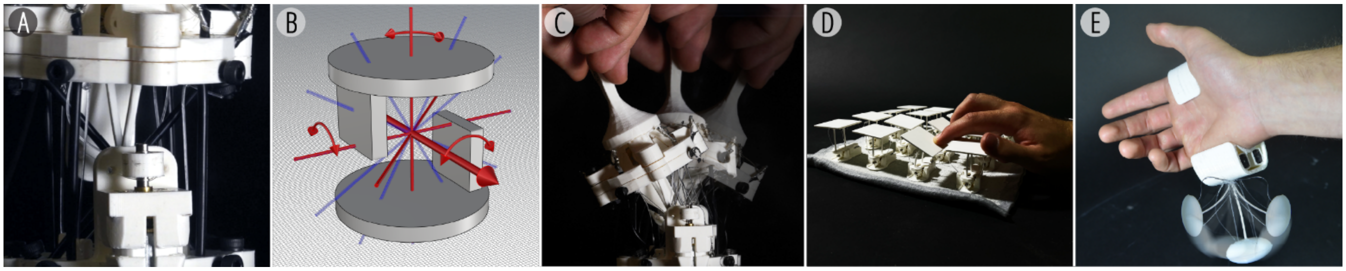


Figure 1: ReCompFig overview - (A) compliant mechanisms that have multiple cable-based reconfigurable kinematic degrees of freedom. (B) Design tool prototype and algorithms based on the screw theory are provided to assist users. Application examples including (C) a multimodal input device, (D) a kinematic material display, and (E) a haptic proxy are provided to demonstrate the enabled interaction design space.

ABSTRACT

From creating input devices to rendering tangible information, the field of HCI is interested in using kinematic mechanisms to create human-computer interfaces. Yet, due to fabrication and design challenges, it is often difficult to create kinematic devices that are compact and have multiple reconfigurable motional degrees of freedom (DOFs) depending on the interaction scenarios. In this work, we combine compliant mechanisms (CMs) with tensioning cables to create dynamically reconfigurable kinematic mechanisms. The devices' kinematics (DOFs) is enabled and determined by the layout of bendable rods. The additional cables function as on-demand motion constraints that can dynamically lock or unlock the mechanism's DOFs as they are tightened or loosened. We provide algorithms and a design tool prototype to help users design such kinematic devices. We also demonstrate various HCI use cases including a kinematic haptic display, a haptic proxy, and a multimodal input device.

CHI '22, April 29–May 05, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3502065

CCS CONCEPTS

• **CCS**; • **Human-centered computing**; • **Human computer interaction (HCI)**; • **Interaction devices**; • **Haptic devices**;

KEYWORDS

Haptic proxies, design tool, kinematic devices, wearables, compliant mechanism

ACM Reference Format:

Humphrey Yang, Tate Johnson, Ke Zhong, Dinesh K. Patel, Gina Olson, Carmel Majidi, Mohammad Islam, and Lining Yao. 2022. ReCompFig: Designing Dynamically Reconfigurable Kinematic Devices Using Compliant

Mechanisms and Tensioning Cables. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29–May 05, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3502065

1 INTRODUCTION

The human hand is a versatile instrument that affords us to tangibly and dynamically interact with the environment. The resulting kinematic experiences make up an essential part of perceiving the world around us. In HCI, we are interested in creating devices that afford these kinematic interactions to support a more natural, intuitive, and enriched interaction experience. Along this line of effort, prior works had explored creating haptic proxies for virtual reality [31], displaying material properties and enabling tangible interactions [5], and designing gadgets and toys [8,9]. However, as we push against the boundary of kinematic device design, creating multimodal (i.e., having more than one kinematic mode) and reconfigurable kinematic devices remains a challenge. Specifically, a mechanical joint (e.g., hinge, slider) is needed for each targeted degree of freedom (DOF), and their integration may require expert knowledge and skills. The footprints of the mechanical components also impose a size limit, making it difficult to create miniaturized and compact interactive devices. Moreover, adding reconfigurability to the devices also requires latching or locking mechanisms, further exacerbating the design and fabrication complexity. As a result, existing interactive kinematic devices are often highly specialized, only possess a single kinematic mode, and cannot adapt nor reconfigure to different use scenarios.

We address these challenges by combining compliant mechanisms (CMs) and tensioning cables to create multimodal and reconfigurable kinematic mechanisms (ReCompFig), as seen in Fig. 1A. Compliant mechanisms are prescribed with DOFs and kinematic behaviors by making local part geometries slender and flexible (i.e., flexural elements) [7]. Compared to conventional joints made of hinges and sliders, CMs are mechanically simple to design and fabricate as it involves fewer parts and thus less assembly. A CM can also be designed to offer multiple DOFs without increasing its mechanical complexity, making it a viable method to design compact and multimodal kinematic devices. However, conventional CMs cannot be reconfigured once fabricated. Therefore, we introduce tensioning cables into CMs as a way to reconfigure (i.e., temporarily lock/unlock individual) DOFs during use as well as add stretchable strain sensors to augment their functionalities as input devices, enabling dynamically reconfigurable kinematic behaviors for HCI uses such as different haptic feedback and physical input modes.

In this work, we leverage and adopt recent advances in CM design methods [6] to propose a framework to assist HCI researchers and practitioners in the design of multimodal and reconfigurable kinematic devices. Our method uses the screw theory to help users achieve two design goals - prescribing desired DOF modes and enabling dynamic DOF tuning. The designed devices would then afford the DOF modes as specified by the user, and the individual modes can be enabled/disabled by selectively tightening/loosening the cables. We also developed the method into a computational tool to assist users in designing these kinematic mechanisms (Figure 1B). Finally, we present application examples including a multimodal input device (Figure 1C), a kinematic material display (Figure 1D), and a miniaturized haptic proxy (Figure 1E) to demonstrate the proposed kinematic mechanisms' relevance to the HCI community and the enabled design space.

While both CMs and tensioning cables are not novel concepts in HCI, the combination is. This work provides tools and algorithms to enable their integration. It is worth noting that our method is focused on designing the *kinematics* of the devices (i.e., how can it move), and their *kinetics* (i.e., how much can it move) is omitted and will be a future research opportunity. Therefore, our primary contributions include:

- Method and principles to design multimodal and reconfigurable kinematic mechanisms.
- Technical artifacts including a design tool and computational algorithms related to multimodal kinematics design.

In addition to the methods and tools, this paper also presents studies and data as a secondary contribution:

- Evaluation of the designed kinematic mechanisms' viability.
- Application examples of multimodal and reconfigurable kinematic devices.

2 RELATED WORK

2.1 Tangible and Kinematic Feedback in HCI

In HCI, we are concerned with recreating different types of haptic feedback to achieve more immersive interaction experiences. Examples of this haptic feedback include weight or volume change [26, 40], forces [22, 32], vibrations [32], texture [19], and input kinematics [1, 30]. In particular, inForm [5] is an instrumentalized table that uses linearly actuated pins to render shapes [11, 45] or kinematic feedback [5, 17, 24] on demand in a range of interaction contexts (e.g., teleconferencing, video games, and composing). ReCompFig builds on top of these works and navigates a larger kinematic interaction design space (i.e., enabling linear and rotational motions in all directions). The DOF reconfigurability also enables us to recreate the haptic sensation of different materials (e.g., liquids, elastic rods, rigid objects) using a single device, making it possible to make compact and portable devices.

2.2 Haptic Proxies in Virtual reality

On the other hand, rendering haptic feedback is also a central topic in virtual or augmented reality. In addition to visual and aural cues, prior works had also explored simulating shapes [4], weight [18], or using a combination of both to imitate the handfeel of grasping different objects [3, 14, 15, 35, 51]. Several motor- and tendon-driven devices were also developed for this domain [2, 33]. This work takes a similar technical approach by using motors and cables. Yet, unlike the predecessors, we use cables to tune the devices' kinematic responses instead of directly providing force feedback. Noticeably, recent developments in this area [31, 42] also started to explore using deformable materials and structures to create force feedback. ElaStick [31], for one, is a reconfigurable device that uses rubber bands and cables to simulate the haptic feedback of swinging a rigid or elastic stick. Yet, as the authors pointed out, the device relied on a pre-designed mechanical structure and had limited degrees of freedom. The mechanical structure based on linkage systems and motors also led to increased weight and size. By contrast, the motions of ReCompFig devices are solely enabled by the structure's flexibility and may enable us to create more compact and lightweight devices. Our design tool and framework also support users to customize devices with the desired DOFs and add sensing capabilities, further expanding the haptic proxy design space.

2.3 Compliant Mechanisms in HCI

HCI researchers have explored using complicated mechanisms to create interactive devices with prescribed kinematic behaviors. Their applications include customized action figures [36], mechanical computers [10], or gadgets and toys [20]. Authoring tools like KinetiX [27] and Metamaterial Mechanisms [8, 10] allowed users to construct 2D, morphing lattice grids by connecting material voxels with prescribed transformation behaviors. In the presence of an external load, the cells would move in synchrony and carry out certain motions. On the other hand, there are also design tools focused on the inverse design of compliant mechanisms. That is, given a targeted motion path [9, 20], deformability [36], or DOFs [34], the design tool will automatically generate a corresponding compliant mechanism for the user. ReCompFig enriches the existing design toolbox by providing a framework and design tool to create 3D and reconfigurable compliant mechanisms. Further, ReCompFig may be combined with transformable robots [13, 23, 39] to create more compact devices that afford versatile kinematic functions or be integrated into mechanical metamaterial designs [8, 10]. When used along with smart materials, ReCompFig may also enable users to design shape-changing interfaces [25, 46, 49, 50] that have context-dependent and complex kinematic behaviors.

2.4 Compliant Mechanism Design Methods

In this work, we take inspiration from the freedom and constraint topology (FACT) design method [6, 37] and adapt it to develop our algorithms and design tools. Compared to other CM methods like topological optimization [16] or pseudo rigid body replacement [41, 44], FACT offers unique benefits, including computational efficiency, and does not require a pre-designed linkage model as input, thus allowing users to interactively and freely explore design opportunities and making it an ideal computational engine. ReCompFig also makes contributions to this domain by proposing novel design algorithms and principles. Specifically, while the FACT method was originally developed to analyze existing compliant mechanisms, its mathematical principles are adapted to create assistive design functions. We also introduce reconfigurability and multimodal kinematics as new dimensions of compliant mechanism design. On the other hand, while this work focuses on designing CM kinematics, it is also possible to program their force-deflection profiles as shown in [41]. Yet, these kinetic responses may require a physically based simulator as a backend design engine.

3 BACKGROUND: KINEMATICS AND COMPLIANT MECHANISMS

3.1 Kinematics in 3D Space

This paper evaluates and designs kinematic devices through their motional *degree of freedom* (DOF) and *constraint* (DOC). A rigid body's configuration in 3D space can be defined by six numbers - their position and rotation along the three principal axes (Figure 2A). A mechanical system is considered to have a kinematic DOF when it is free to translate along or rotate about an axis with a small force. Similarly, a system is deemed to have a degree of constraint when it generates negligible motion along or about an axis. A device's DOFs and DOCs are also complementary: DOF+DOC=6 motional freedoms in 3D space. These DOFs and DOCs can be represented by screw vectors (see also the Algorithm section). Given a system, any linear combinations of its DOFs or DOCs are also valid and would generate new motions or constraints, and the collection of all linear combinations are called *freedom* or *constraint spaces*, respectively.

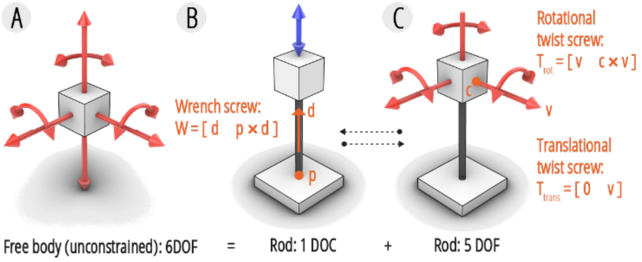


Figure 2: The screw theory of motion - (A) a rigid body's six motional degrees of freedom in a 3D space, (B) flexural rods as wrench screw vectors, and (C) motions as twist screw vectors.

3.2 Compliant Mechanisms

Compliant mechanisms [7] are mechanical structures that enable motion by the flexibility of materials. These structures can be made monolithic (i.e., single-pieced) and are widely used across different scales and domains: from microelectronics [12] to architecture [29]. A CM comprises slender flexural elements connecting between rigid bodies, and its DOFs are determined by the flexures' layout (i.e., positions and directions). Structurally speaking, flexures cannot buckle nor extend when subjected to axial loads, but they can bend under lateral forces or bending moments. It is worth noting that based on this formulation, a CM's motional freedom is only concerned with the orientation and placement of the flexures, and their structural properties (e.g., diameter, stiffness) have no effect on its DOFs. However, these structural properties may affect the compliance and stiffness of CMs [43]. Similarly, as long as they are sufficiently stiff, the rigid stages' shape would not affect the CM's kinematics.

4 DESIGN PRINCIPLES

4.1 Expanding Flexure Design Space Using Cables

ReCompFig comprises three building blocks: rod flexure for rigid support and permanent constraints, tensioning cable for DOF/DOC reconfigurability, and elastic cables for sensing.

4.1.1 Rod flexure. In CMs, a rod flexure typically has an aspect ratio between 1:10 to 1:20. It has buckling strengths well above the expected use-load and negligible tensile extensions in normal situations, meaning they cannot buckle nor extend under normal use

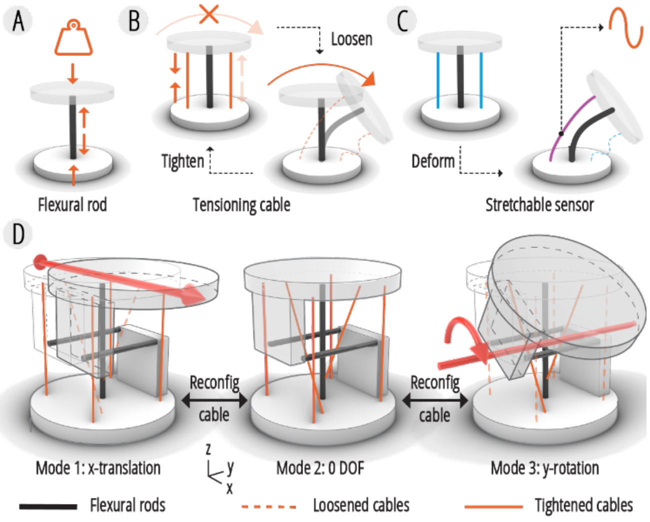


Figure 3: Design principles of reconfigurable, sensing, and multimodal kinematic devices. (A) A flexural rod constrains DOFs by resisting compression and extension, whereas (B) a cable only constrains extension when tightened, and (C) an elastic cable (sensor) imposes no constraint. (D) The tensioning cables can be tightened/loosened to reconfigure the device's degree of freedom.

scenarios, therefore limiting linear motions along its longitudinal axis (Figure 3A).

4.1.2 Tensioning cables. Inelastic tensioning cables also hold similar structural properties: they have negligible extension when subjected to tensile loads. Yet, unlike rod flexures, cables would buckle when subjected to compressive forces due to their extreme aspect ratio (typically larger than 1:40) and low buckling strength, therefore making them "half rods" in a structural sense, i.e., cables behave like rods under tensile loads but are virtually ineffective under compression (Figure 3B). Nonetheless, we can still simulate a complete rod flexure by placing a pair of cables on both sides of the motional axis, such that when one buckles under compression, the other withstands the forces in tension. Based on this property, we leverage cables as an on-demand, reconfigurable flexure element to design CMs. When controlled by a motor, the cable pair can be further shifted in or out of effect when tightened or loosened, respectively, thus enabling reconfiguration for different interaction scenarios without modifying its structure (Figure 3D).

4.1.3 Elastic sensing cables. On the other hand, cables made of elastic materials hold no structural functions in a CM because their elasticity allows them to stretch under tension, thus cannot be used as flexures in any way. However, we can take advantage of this fact and add stretchable sensors (e.g., conductive rubber cord stretch sensors) to detect its deformations without modifying its DOFs (Figure 3C). This addition allows us to identify how the user interacts with the kinematic mechanism and use it as an input device.

4.2 Designing Multimodal Kinematic Mechanisms

The cable-driven reconfiguration can be used to create devices with dynamic and distinct kinematic modes and affordances. These devices can be analytically designed using the screw theory and a Venn diagram of constraint/freedom spaces in several steps (Figure 18). First, we can identify their corresponding constraint spaces given the desired DOFs for each kinematic mode. We can then use these constraint spaces to determine the placement of the non-reconfigurable rods - the flexural rods shared by all kinematic modes. At this point, the mechanism should have the combined DOFs of all kinematic modes. Next, tensioning cables are added for each kinematic mode to constrain undesired DOFs, and a control scheme (i.e., a table that matches cable group status with kinematic modes) is generated. Stretchable sensors are also optionally added to the device following the tensioning cables. Finally, the model is completed by assigning a thickness to the rods and designing guiding tubes for cables and housings for motors.

In the following section, we will demonstrate the design process of a multimodal input device by following these steps. The process is assisted by a design tool prototype and uses the screw theory and the FACT method as a back-end engine. The tool takes kinematic mode specifications as input and provides prompts and visualizations that guide users through the design steps. Note that instead of generating a design solution for the user, the tool suggests where the flexures and cables can be placed, and the user should place the elements based on the prompts. This modality of assistance allows the user to make informed design decisions while leaving the room for creativity and freedom to achieve different design goals (e.g., aesthetics, functional considerations). The algorithms behind the design tool and physical implementation are also explained in later sections.

5 RECOMPFIG WALKTHROUGH: MULTIMODAL INPUT DEVICE

This section describes the user workflow through the design of a multimodal input device (Figure 4). Figure 5 through figure 11 show the design tool and process with slight adjustments to colors and fonts to promote readability. Yet, the text overlays on the model view are superimposed to explain the visualizations provided by the tool. The device has three kinematic modes - slider, joystick, and dial knob - that provide different interaction affordances, and the goal is to produce all these functionalities within a single CM joint. The algorithms are implemented in Python, and the tool in Rhinoceros 3D with plugins (Grasshopper, Human UI, and CPython), and the three enabled interaction modes are described later in Figure 12

5.1 Step 1: Assign DOFs to Each Kinematic Mode

The workflow starts with modeling the two rigid stages - a fixed and a free end - of the kinematic device (Figure 4A), which were left plain since the cables, motor housings, and flexural rods will be added in later steps. The user begins by creating three DOF modes for this device (Figure 5). The first mode approximates a joystick, which allows for rotations about any axis on the x-y plane

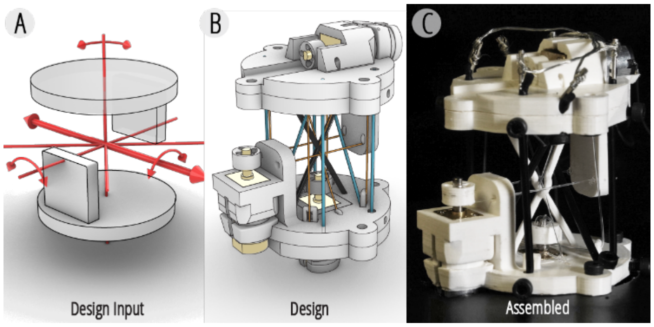
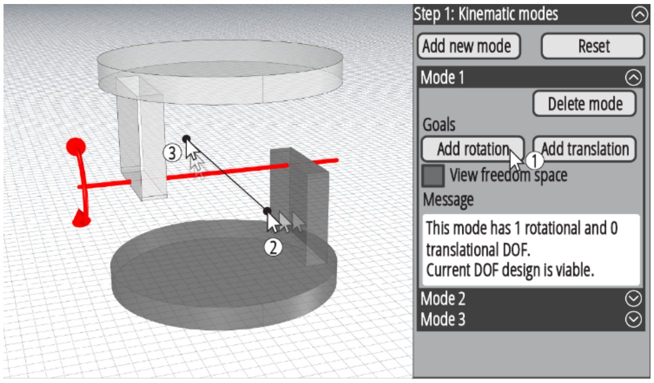


Figure 4: A multimodal input device designed using the proposed framework - (A) the design goals and (B) the resulting model, and (C) the assembled device.



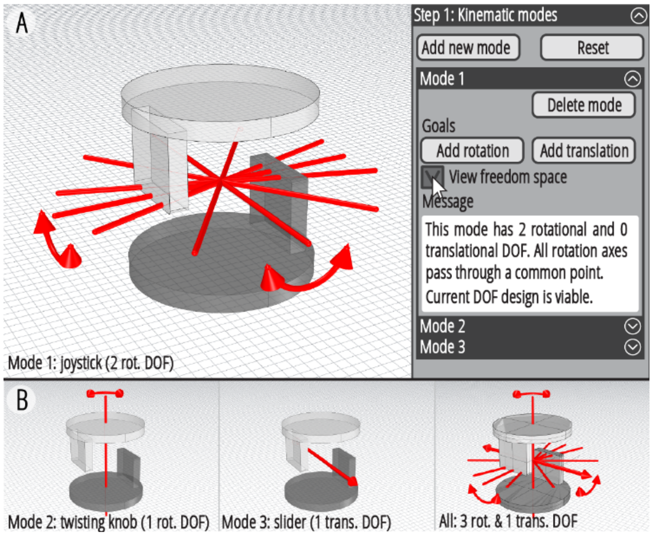
This mode has 1 rotational and 1 translational DOF.
Current DOF design is viable.

Figure 5: After importing the input model, the user begins the design task by adding kinematic modes and specifying the DOFs of each mode.

that passes through the center of the mechanism. To represent these freedoms, the user adds several possible rotation axes to the design tool, which in return visualizes the freedom space for the user (Figure 6A). Similarly, the user assigns a z-axis rotation to the second kinematic mode to simulate a dial knob and a translational motion along the x-axis to the third mode as a slider (Figure 6B).

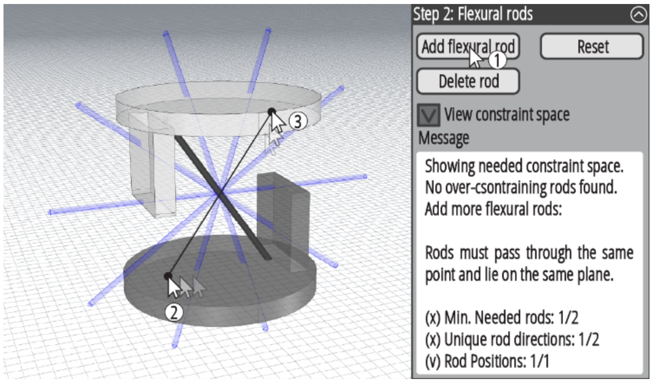
5.2 Step 2: Place Flexural Rods

Once the user specifies the DOFs of all kinematic modes, the design tool then proceeds to guide the user through designing the flexural rods. These rods are shared by all kinematic modes and are not reconfigurable. The constraint space visualization suggests the location and orientation of permitted rod placements. The user can take this information and the textual descriptions to place the flexures iteratively until the constraint spaces are satisfied and complete (Figure 7). In this case, the tool prompts the user to add at least two rods that pass through the center point and lie on the y-z plane. If the user places a rod in an invalid position or orientation, the CM will be over- and ill-constrained, and the design tool will highlight and prompt the user to correct it (Figure 8).



This mode has 2 rotational and 0 translational DOF. All rotation axes pass through a common point.
Current DOF design is viable.

Figure 6: (A) Once the DOFs are assigned, the design tool provides visual and textual prompts to inform the user of the motional freedom space. (B) The DOF goals added for each kinematic mode.



Showing needed constraint space.
No over-csontraining rods found.
Add more flexural rods:

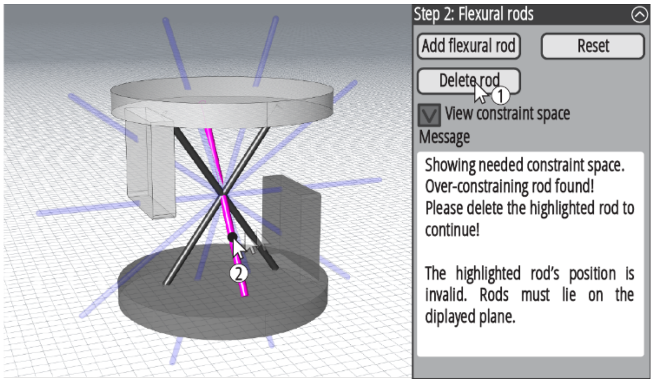
Rods must pass through the same point and lie on the same plane.

(x) Min. Needed rods: 1/2
(x) Unique rod directions: 1/2
(v) Rod Positions: 1/1

Figure 7: Once the desired DOFs are specified, the design tool then prompts the user to add flexural rods that are shared by all kinematic modes.

5.3 Step 3: Place Tensioning Cables

Following placing the flexural rods, the design tool guides the user to place tensioning cables that reconfigure the kinematic device (Figure 9). The tensioning cables are assigned into groups that are actuated together, which also produces a reconfiguration plan for the kinematic modes. The design tool visualizes the cable's permissible placements to guide the user through placing the cables. A highlight also guides the user to place the cables in the proper orientation. Since the user should add cables in pairs to balance their loads, the design tool will generate the coerced twin-cable whenever the user places one. The tool also prompts the user when the cable placement is invalid. All cable groups must be satisfied to proceed to later steps.



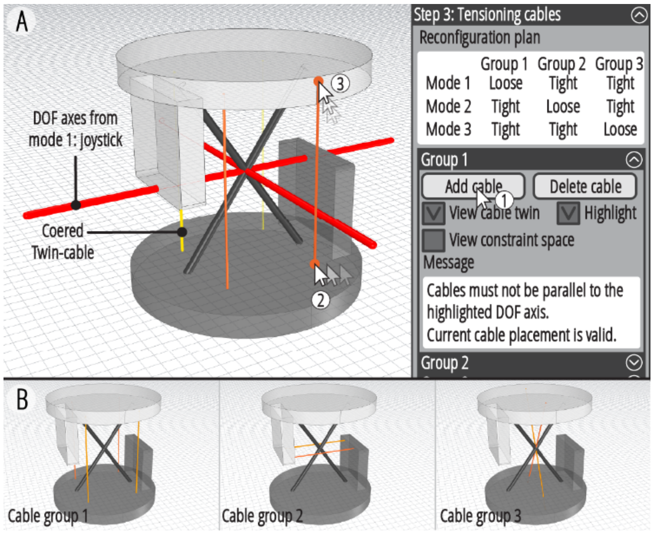
☑ View constraint space

Message

Showing needed constraint space.
Over-constraining rod found!
Please delete the highlighted rod to continue!

The highlighted rod's position is invalid. Rods must lie on the diplayed plane.

Figure 8: If the user adds a flexural rod in an invalid placement, the tool will highlight the rod and inform the user to remove it.



Reconfiguration plan

	Group 1	Group 2	Group 3
Mode 1	Loose	Tight	Tight
Mode 2	Tight	Loose	Tight
Mode 3	Tight	Tight	Loose

☑ View cable twin ☑ Highlight
☐ View constraint space

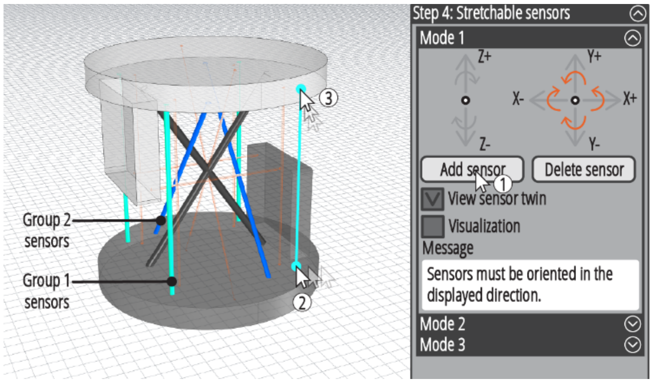
Message

Cables must not be parallel to the highlighted DOF axis.
Current cable placement is valid.

Figure 9: (A) The design tool provides visual and textual prompts to inform the user of valid tensioning cable placements. (B) The tensioning cable groups added for this design.

5.4 Step 4: Place Sensors (Optional)

The user has the option to add stretchable sensors to the CM to convert it into an input device, and the design tool also guides the user to place the sensors in appropriate orientations. The rules and textual prompts are similar to that of the tensioning cables. However, the design tool provides a panel to preview which DOF's motion is detectable given the current setup (Figure 10). The design tool also tells the user which sensors are stretched along and responsive to a DOF, informing the circuit logic and controller firmware design. Figure 10 shows the user adding pairs of sensors to detect the deformations of each kinematic mode.

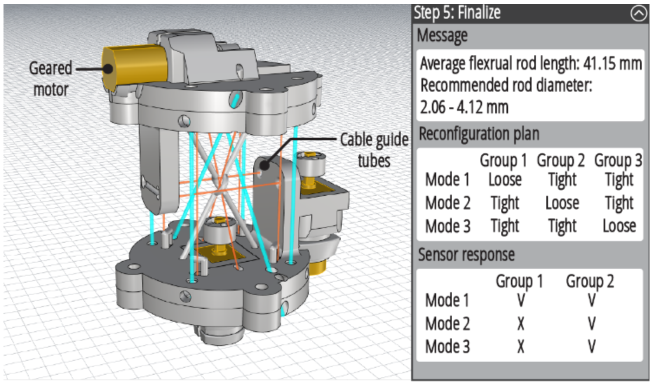


☑ View sensor twin
☐ Visualization

Message

Sensors must be oriented in the displayed direction.

Figure 10: The user has the option to add stretchable sensors to the kinematic device, and the design tool checks if a DOF can be sensed given the current sensor layout.



Message

Average flexrual rod length: 41.15 mm
Recommended rod diameter:
2.06 - 4.12 mm

Reconfiguration plan

	Group 1	Group 2	Group 3
Mode 1	Loose	Tight	Tight
Mode 2	Tight	Loose	Tight
Mode 3	Tight	Tight	Loose

Sensor response

	Group 1	Group 2
Mode 1	V	V
Mode 2	X	V
Mode 3	X	V

Figure 11: Once a design is completed, the user proceeds to finalize the design by adding mechanical details and modularizing the model for fabrication.

5.5 Step 5: Model Flexure Elements and Rigid Stages

Once the kinematic mechanism is completed, the user should finalize the model by adding mechanical components and assigning thicknesses to the flexural rods (Figure 11). The rods have a diameter of 2 mm, which is approximately 5% of their length, to ensure sufficient deformability. The user has the freedom to modify the rigid stages into any shapes given they are sufficiently stiff. In this design, the user added guiding tubes and anchors for the cables, housings for the geared motors, and insertion holes. The mechanism was also divided into four parts for 3D printing.

6 APPLICATION EXAMPLES

6.1 Multimodal Input Device

Graphical user interfaces on a screen can change from one mode to another in split seconds. E.g., a slider can change into a knob with virtually no delay. However, unlike their digital counterparts, physical input devices like keyboards, mice, and joysticks only have a single input mode. Users are often required to switch between

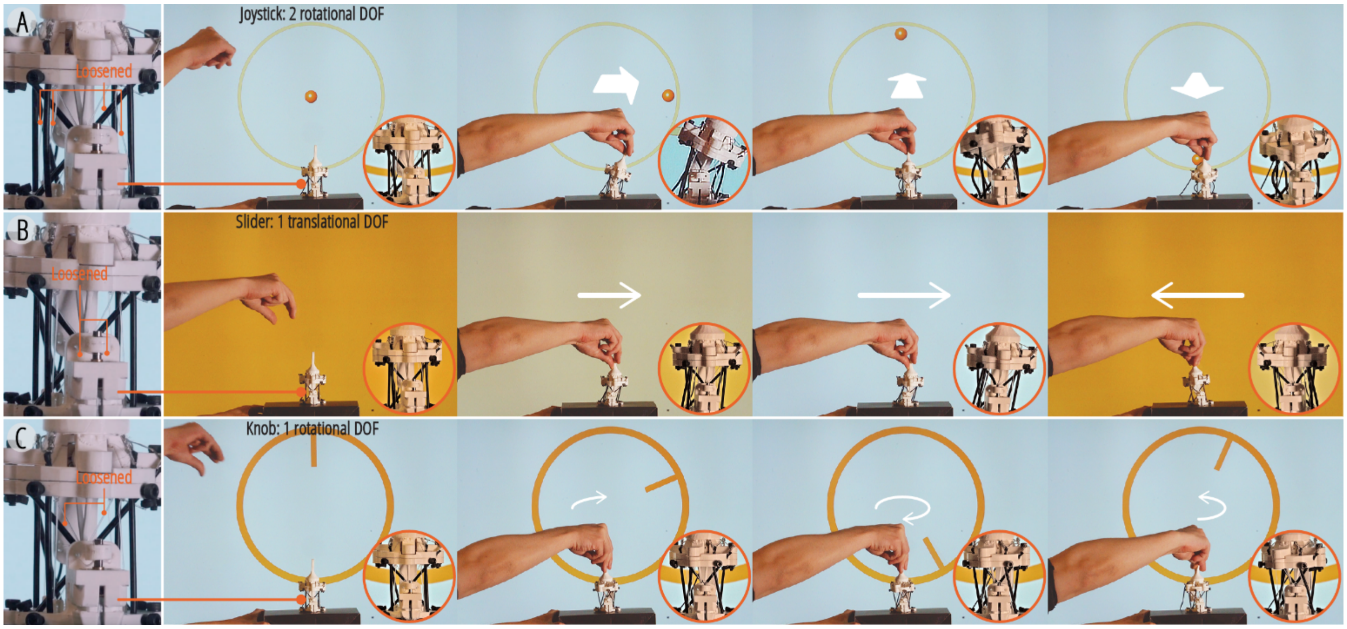


Figure 12: The multimodal input device functions as (A) a joystick to move a ball on the screen in the background, (B) a slider controlling the color on the screen, and (C) a twisting knob to rotate a digital dial. The leftmost images show the cable status under each mode (blue: loosened, orange: tightened), and the insets show zoomed-in views of the device.

them for different interaction scenarios. Here, we demonstrate that as a multimodal interface, a ReCompFig input device can support various kinematic interactions at a time (Figure 12). The device can change between three kinematic modes on demand, each recreating the haptic experience of a familiar interface (i.e., joystick, slider, twisting knob).

This design has three groups of tensioning cables and six stretchable sensors to detect the user's action. The reconfiguration takes less than a second, achieving almost real-time modality change. However, based on our user experience, we report that while the device is qualitatively robust and supports large motion ranges (e.g., >45° of rotation), it requires a relatively larger force to activate than conventional input devices. Future work may consider optimizing the flexure dimension or choosing a more deformable flexure rod material to achieve a more comfortable interaction. On the other hand, we also speculate that taking advantage of the device's mechanical simplicity, it might be possible to miniaturize the design to embed it into commercial products (e.g., video game controllers) and enable more dynamic interaction experiences.

6.2 Kinematic Material Display

In this example, we leverage the compactness and reconfigurability of ReCompFig mechanisms to create a kinematic material device. Different from conventional displays that render images or shape displays that physicalize geometries, this kinematic display is used to tangibilize the kinematic freedoms of a piece of material. I.e., an object's deformability when touched by hands. The display comprises a 4x4 grid of individually addressable kinematic bits (Figure 13A). The modules have a pair of cables to enable/disable their translational and rotational DOFs (Figure 13B). Figure 14 shows the device in action: when the cables are loosened, the interface has the kinematic affordance of mud, whereas when tightened, the display simulates stiff dried soil.

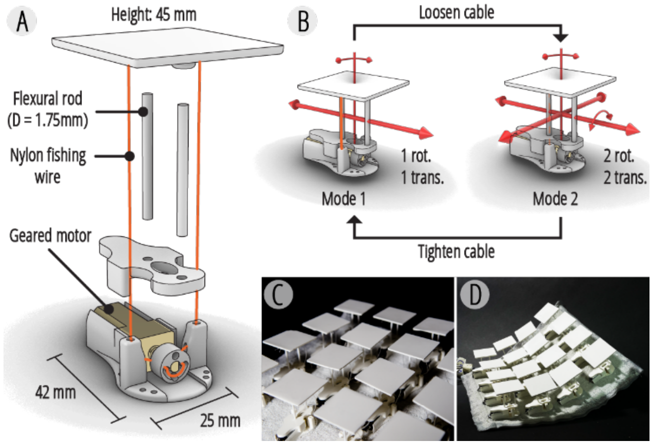


Figure 13: Kinematic material display - (A) unit module design and (B) DOFs before and after cable tightening. (C) Arrays of kinematic modules as an array-like display. (D) The display conforming to a curved surface.

In this application example, we report that qualitatively speaking, designing kinematic devices using ReCompFig brings about several advantages. Compared to the inForm table [5] that provides kinematic response by actuating linear pins, our device affords more kinematic motions (bending, twisting) while having a smaller

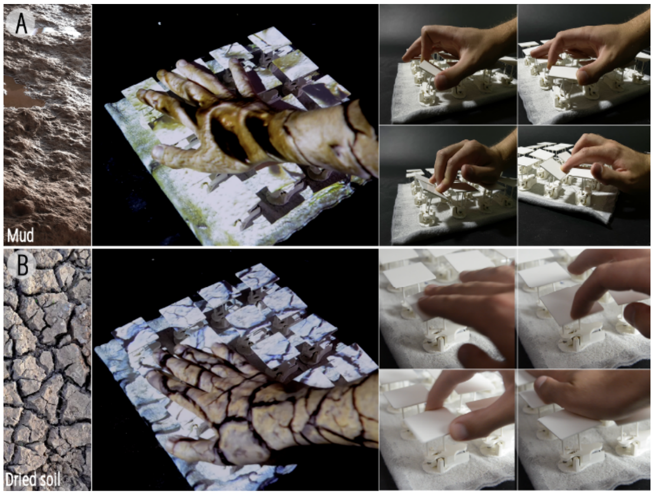


Figure 14: Interacting with the kinematic material display, simulating (A) mud with four DOF and (B) stiff dried soil with 2 DOF.

footprint. When sewed onto a fabric substrate, the display can even wrap around nonplanar surfaces (Figure 13D). Future work may also consider incorporating actuatable tendons to further instrumentalize this design.

6.3 Wearable Haptic Proxy

We demonstrate that ReCompFig mechanisms can also be used to create kinematic haptic proxies that simulate the handfeel of holding objects made of different materials (Figure 15A), similar to that of ElaStick [31]. The device is attached with a weight of 80 grams, which can swing in different directions depending on the tensioning cable configuration (Figure 15B). Specifically, the weight is immobile when all cables are tightened and free to rotate or translate along five DOFs when fully unlocked. Partial cable tightening simulates unidirectional weight shift along a single axis. This design allows us to simulate the weight shift of holding different objects, including liquids, elastic sticks, and rigid bars (Figure 16A, B, and C, respectively). It is worth noting that the mechanical simplicity of our design led to a substantial reduction of assembly demand and device weight: it consists of only three printed parts, two mini-motors, and weighs 101.3 g, whereas the device was designed with more than four parts, four motors, and weighs 814.4 g in ElaStick [31]. For this reason, we hypothesize that ReCompFig can enable us to create lightweight haptic proxies. Moreover, the device's design frees the user's fingers to interact with other possible interfaces (e.g., texture, vibration), or to hold a representative prop object, making room for a more immersive experience. While not explored in this work, we believe future work can also integrate this design with weight-changing interfaces [26] to become a more versatile and immersive haptic proxy in virtual or mixed reality.

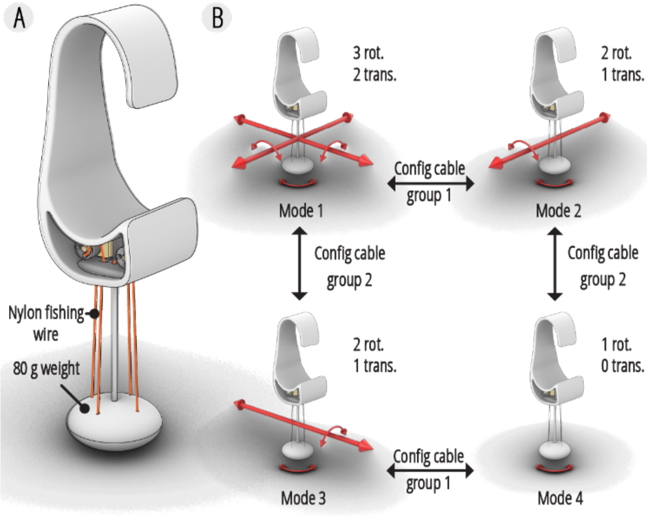


Figure 15: Kinematic haptic proxy - (A) its structure and (B) kinematic modes controlled by a pair of motors.

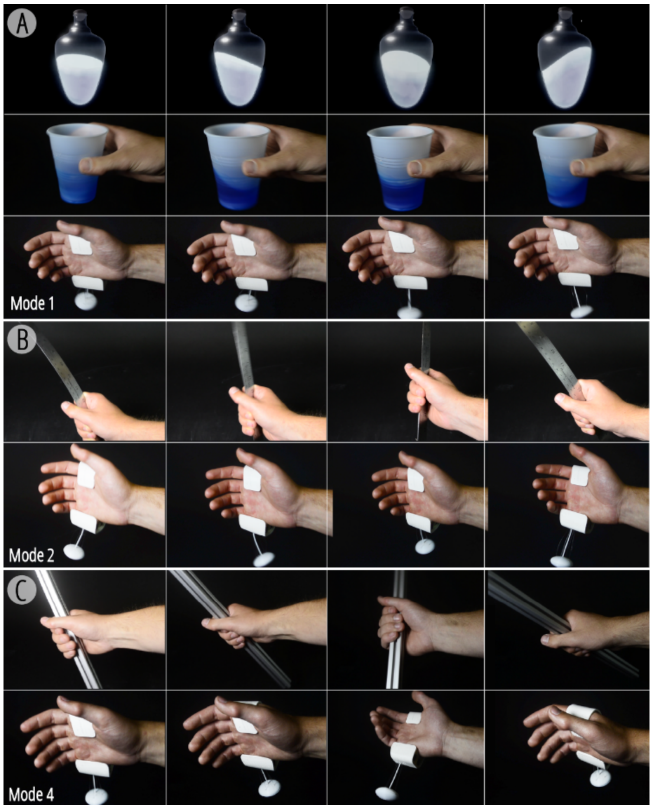


Figure 16: Simulating the haptic feedback of (A) liquids, (B) elastic sticks, and (C) rigid bars with the kinematic haptic proxy.

7 ALGORITHM

This section will first review the fundamentals of the screw theory on which our algorithms are based. Next, in the ReCompFig Design Algorithms, we will elaborate on our adaptation of the screw theory to design multimodal and cable-reconfigurable kinematic mechanisms.

7.1 Screw Theory Basics

Screw theory [38] is a generalized geometric formulation of motional DOFs. A DOF is represented by a six-dimensional screw vector that encodes the direction and position of the translational/rotational axis. A screw vector T (also called a twist vector) is defined as:

$$T = [\boldsymbol{v} \;\; c \times v + n \cdot v] \tag{1}$$

in which v and c are 3D vectors representing the direction of and a point on the axis, respectively, and n is a scalar representing the pitch (i.e., the ratio between translation and rotation) of the screw motion (Figure 2C). These vectors can also be linearly combined into a new motion, and the linear span of a mechanism’s DOFs $[T]$ is called its freedom space. In this paper, we are mainly focused on pure translations and rotations. For pure rotations, $n = 0$ and v is a non-zero vector; for pure translations, they are assumed to have an infinite pitch, and the vector has a normalized form of $T = [0 \;\; v]$. We refer readers to [6, 38] for more details.

The CM flexures and their corresponding DOCs can also be modeled using the screw theory. Take the rod flexures used in the paper for example (Figure 2B), their constraining screw W - also called a wrench vector - can be represented using a scheme similar to (Eq. 1):

$$W = [d \;\; p \times d] \tag{2}$$

where d is the longitudinal axis of the rod and p is a point on the rod. The two parts of the wrench vector are also called the directional and positional components, respectively. Like the DOFs, a CM’s constraining screws can also be linearly combined to form constraint spaces $[W]$. Since the constraints and freedoms are complimentary (Figure 17), they can be solved analytically when the other is known [37]. Given a CM consisting of multiple flexures:

$$[W]\,\Delta[T]^T = [0] \tag{3}$$

where $[W]$ and $[T]$ are the system’s constraint and freedom spaces, and Δ is a swap operator defined in [37], and $[0]$ is a 6×6 zero matrix. A CM device design is considered complete when (Eq. 3) is satisfied.

Two algebraic concepts are frequently used in designing CMs: *nullspace* (also called the *kernel*) and *rank*. The *nullspace* of a subspace is identical to a NOT operator in Boolean algebra. When applied to freedom subspaces, it is identical to finding a system’s immobile motions. Similarly, a *nullspace* is the unallowed flexural rod placements when applied to constraint subspaces. On the other hand, the *rank* of a subspace shows how many principal vectors (i.e., unique screw vector directions and/or positions) are needed to recreate a system or a linear subspace.

7.2 ReCompFig Design Algorithms

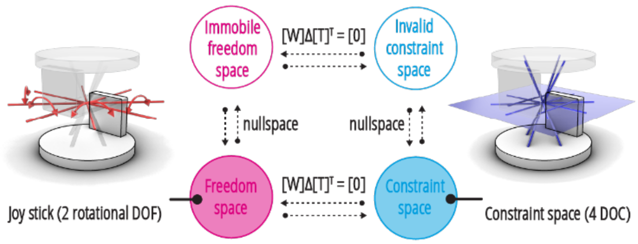


Figure 17: The freedom and constraint topology of a compliant mechanism. The spaces are linear systems (circles in the figure) with one-to-one algebraic mappings (annotated in dashed lines and arrows). We can visualize the space into 3D geometric entities (models on both sides).

7.2.1 *Constraint and Freedom Spaces of Kinematic Modes.*

The constraint and freedom space of each kinematic mode can be calculated based on the motion axis specified by the user in step 1. The line itself creates a directional vector, and either of the endpoints can be used as a reference point on the axis. Plugging these components into (Eq. 1) yields a twist vector of the desired motion. However, since the motions assigned under a kinematic mode may contain linear redundancies that may complicate the calculation, it is important to simplify the freedom space before computing its complementary constraint space. Given a matrix $[T_i^{'}]$ that collects all twist vectors under a kinematic mode i, its non-redundant freedom space $[T_i]$ can be found by finding the *kernel* of its *nullspace*. The outcome $[T_i]$ can then be used with (Eq. 3) to compute the kinematic mode’s constraint space $[W_i]$.

7.2.2 *Shared Flexure Placements.*

Given a collection of constraint spaces associated with each kinematic mode, the shared flexure placements $[W_{shared}]$ can be identified by finding their intersection (Figure 18A). I.e., the wrench vectors shared by all modal constraint spaces. This is achieved by deriving a matrix $[T_{all}]$ that contains all twist vectors under all kinematic modes and finding its corresponding constraint space using (Eq. 3). A device design is considered unviable if $[W_{shared}]$ is empty or has no rod-directional components.

7.2.3 *Detecting Over-Constraining Elements.*

If the current design’s axial or positional component’s *rank* is higher than that of the target, then the system is considered over-constraining, and the tool should prompt the user to reduce the degree of constraints by removing the respective flexural elements. Over-constraining flexural rods can be identified by checking whether they are linearly spanned by the target constraint space (i.e., a subset or subspace). Similarly, if the current design’s *rank* is lower than that of the target, the system is under-constrained, and the tool will prompt the user to add more flexures.

7.2.4 *Tensioning Cable.*

The tensioning cable groups can be identified using a Venn diagram (Figure 18B). Each subset (subspace) in the diagram corresponds to a constraint space $[W_{cable}]$ that needs to be satisfied through a cable group, which is the difference between $[W_{cable\ shared}]$, the intersection of corresponding kinematic modes’ $[W_i]$, and the constraint space $[W_{subset}]$ already

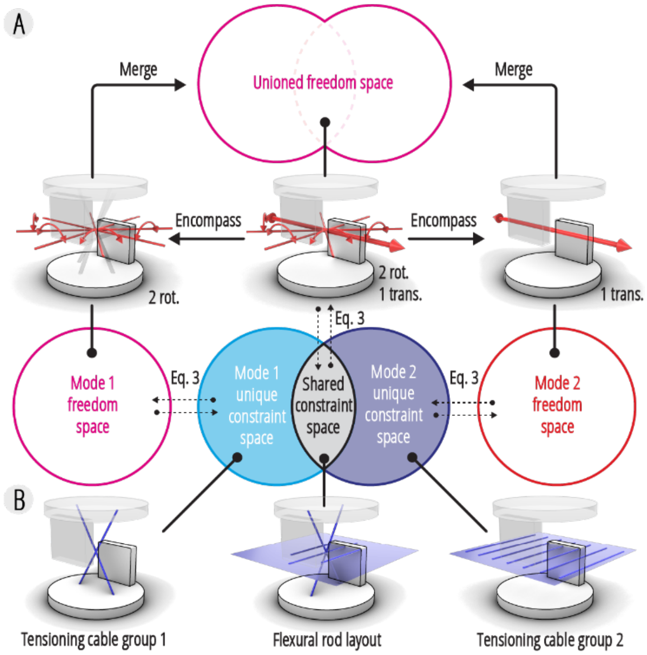


Figure 18: (A) Computing shared flexural rod constraint space - the shared rod placement is computed by intersecting the constraint spaces of all kinematic modes. The dashed lines with arrows denote the algebraic mapping described in Eq. 3. (B) The rods and tensioning cables' constraint subspaces are computed based on the Venn diagram using linear subspace intersection and difference.

satisfied by its subset spaces. For a cable group, $[W_{subset}]$ is found by collecting its subset spaces' $[W_{cable}]$ or $[W_{shared}]$ and taking the *kernel* of its *nullspace*. The group's $[W_{cable}]$ can then be computed as the *kernel* of the concatenation of [W_(cable shared)]'s *nullspace* and $[W_{subset}]$. If a $[W_{cable}]$ ends up empty, then it can be omitted during the design process. Some resulting $[W_{cable}]$ may also have all-zero directional components, which, geometrically speaking, results in invalid placements (i.e., axis-less). In this case, we can take the directional components from $[W_{subset}]$ to generate a valid constraint space.

The cable reconfiguration plans are also generated according to the Venn diagram. To enable a kinematic mode, all of its subset constraint spaces (i.e., cable groups) must be tightened and the rest loosened. Figure 18B exemplifies a case of two kinematic modes, but the method is also applicable to designs with more complex designs. Generally speaking, for translational constraint cables, their directions must not be perpendicular to the translational axis, whereas, for rotational constraints, the cables must not be parallel to the rotational axis nor pass through the rotational axis.

A cable's coerced twin can be found by rotating its wrench vector 180 degrees around the motional axis they are constraining (Figure 19A, B). Additionally, after rotation, the corresponding endpoints in the cable pairs must land on different rigid bodies to ensure the cables experience opposite axial forces under external loads (Figure 19C).

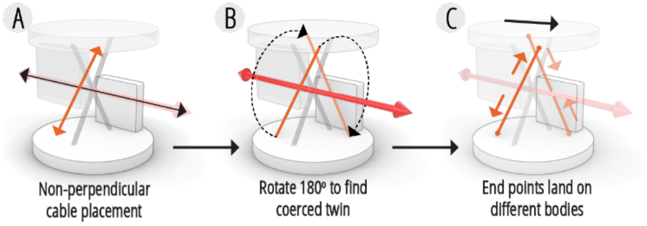

A
B
C
Non-perpendicular cable placement
Rotate 180° to find coerced twin
End points land on different bodies

Figure 19: Tensioning cable pair placement of a translating joint. (A) The cable must not be perpendicular to the translation axis. (B) The coerced twin of a cable can be found by rotating the cable 180 degrees about the translation axis, but (C) the corresponding endpoints must land on a different rigid body, which leads to opposite loads when deformed.

7.2.5 Stretchable Sensor Placement. The sensors have no constraining effects due to their extensibility and can thus be placed freely in the device. Their twins can be found using the same method as the cables. A sensor is responsive to a kinematic mode if its wrench vector is not included (i.e., spanned) by the mode's constraint space $[W_i]$.

7.2.6 Visualization. Conceptually, a twist or wrench vector can be visualized by decomposing them into their building components - axis vectors and reference points on axes. The axis vector is the first half of a screw vector, and the reference point can be found by computing the cross-product of the reversed axis vector and the latter half of the screw vector. The freedom and constraint spaces spanned by multiple screw vectors can also be visualized by linearly combining the screw vectors within that space. We can also pivot our screw space visualization around a point of interest (e.g., the center point of the kinematic device) by moving the reference point along the directions allowed by the space.

7.2.7 Textual Prompt. The textual prompts guide users to design flexure placements that satisfy the constraint spaces. This can be done by comparing the current design with the targeted constraint spaces. The directional and positional components of the constraint space can be evaluated individually, and each is considered complete when the current design's respective screw vector parts have the same *rank* as the targeted constraint space. Finally, the minimally needed and existing number of unique flexural rods can be calculated as the *rank* of the non-redundant constraint spaces.

8 PHYSICAL IMPLEMENTATION

8.1 Flexure Dimension

We acknowledge that there is currently no analytical way to design flexure dimensions. It requires numerical methods like finite element analysis to iterate the design, which is computationally expensive and may render the design process less interactive. Moreover, different application scenarios may also have different structural demands. For this reason, we recommend users iteratively find out the ideal flexural dimensions by physical prototyping and performance evaluation (see Validation section). As a rule of thumb, the flexural rods are recommended to have an aspect ratio (diameter/length) between 0.1 and 0.05. Larger aspect ratios may cause

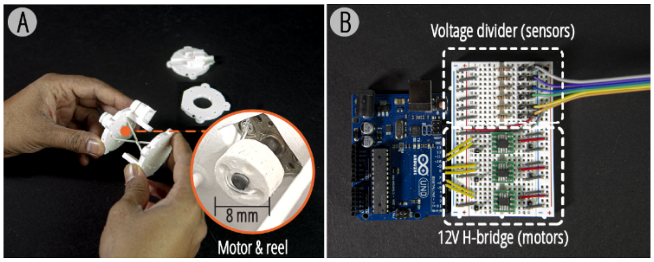


Figure 20: The fabrication and control method using the multimodal input device as an example. (A) The device was printed in parts and assembled together. The image inset shows the reels used to control the cables. (B) The electronics and circuits used to operate the devices.

the rod to become less compliant, whereas smaller ratios may make them more susceptible to buckling but more deformable.

8.2 Fabrication Method

This paper uses a desktop 3D printer (Ultimaker S5) and off-the-shelf filaments (Ultimaker PLA) to fabricate our prototypes. The flexural rods and rigid stages were printed as separate parts and assembled together to form the work prototypes (Figure 20A, Figure 4C). Yet, generally speaking, CMs can be made using any fabrication method and material given adequate resolution and structural properties (e.g., resilience, stiffness, deformability). Other additive manufacturing methods like laser sintering (metal or plastics) or digital light processing (resin) can also be used to fabricate CMs, potentially as a single piece and further reducing the assembly labor. Nylon fishing lines were used for the tensioning cables.

8.3 Control and Sensing

The cables are modulated using geared motors (ROB-12285, SparkFun Electronics) and reel (Figure 13A, Figure 20A), which are controlled by an Arduino board and an H-bridge (BD62130AEFJ-E2, Rohm Semiconductor). Our geared motors have a gear ratio of 298:1, and due to its high reduction ratio, the motor only requires power during reconfiguration, thus making them power-efficient when idle. The cables are threaded continuously through the motor reel to produce an even tightening effect between both cables. We calibrate the cables by manually and procedurally tightening them until the corresponding DOFs become sufficiently stiff. The cables can be modulated during operations by running the motors in the corresponding directions for a set amount of time and travel (i.e., open-loop control). In our case, it takes 0.5 seconds for the motors to reconfigure the cables.

The stretch sensors (product Id: 519) are purchased from Adafruit and have a resistance of 350 ohms per inch and a maximum strain of 70%. The sensor's resistance increases as a function of its strain and can be mapped to a CM's deformation (Figure 20B). Yet, we take a digital approach and detect deformations by checking the resistance deviation from the relaxed state. I.e., a signal is detected if the resistance change exceeds a certain threshold.

9 VALIDATION

9.1 Testing Setup

We evaluate our proposed framework's effectiveness in prescribing DOFs and the cable-driven reconfiguration. Five samples were designed using our design tool and methods. The passive CMs (i.e., A1, B1, and C1) in figure 21A are used to verify the desired DOFs while the CMs with cables (i.e., A2 and B2) are tested for their reconfigurability (Figure 22A, Figure 23A). We use a 3D-printed jig to adapt our CM prototypes to the testing system (Instron 5969) as shown in figure 22C and figure 23C. The samples are fixed on a slider on one end and the loads are applied on the other. The slider ensures the point of force application stays lined up with the machine. A maximum tensile load of 5 N is set for all tests, and the loads were gradually applied at a rate of 5 mm/minute. The test system measures the samples' deformations as their displacement over load. For samples with rotational DOFs, the extensions can be trigonometrically converted into bending angles by plugging in the distance between the load application point and the rotational axis, which is part of the sample's geometry. The tests were repeated five times, and the plots were produced by averaging the results.

9.2 Designing Desired DOFs

The test samples A1 and B2 are designed to validate that the framework can produce the targeted rotational and translational DOF: they are designed with a rotation about the x-axis and a translation along the x-axis, respectively. The mobilities along all six DOFs were evaluated for both samples and can be examined by their deformation over load (i.e., the slope of the curves). A steep curve suggests that the mechanism is less mobile along that direction, whereas a more gradual curve indicates freedom. In figure 21B, C, we can see that both samples are much more compliant along their prescribed motional axes. Compared to their constrained motions, both of the samples had a 5-6x higher deformation under the same load along its DOFs. Specifically, A1 had more than 30 degrees of rotation about its mobile x-axis, whereas the same deformation was almost indiscernible (smaller than 6 degrees) in other directions. Similarly, B1 translated more than 6 mm along its mobile axis while the deformation was less than 0.5 mm in the other directions.

On the other hand, sample C1 is designed with a rotational DOF along the z-axis and a translation DOF along the x-axis, and it was used to validate that the design tool can produce a single mechanism with multiple DOFs. Based on the results, we can see that the sample is much more mobile along the two prescribed DOFs than the others, thus proving that the sample was successfully made mobile in the two targeted DOFs and constrained in the other four. These results show that the proposed framework and design tool can indeed lead to kinematic mechanisms with the desired DOFs.

9.3 Reconfiguration

The test samples A2 (Figure 22A, C) and B2 (Figure 23A, C) are designed with the same DOFs and flexural rod layout as their counterparts in the earlier experiments, A1 and B2, respectively. However, their DOFs can be enabled or disabled by loosening or tightening the cables placed per the design tool's suggestions. Figure 22B shows that under a five Newton load, the maximum rotational angle of A2

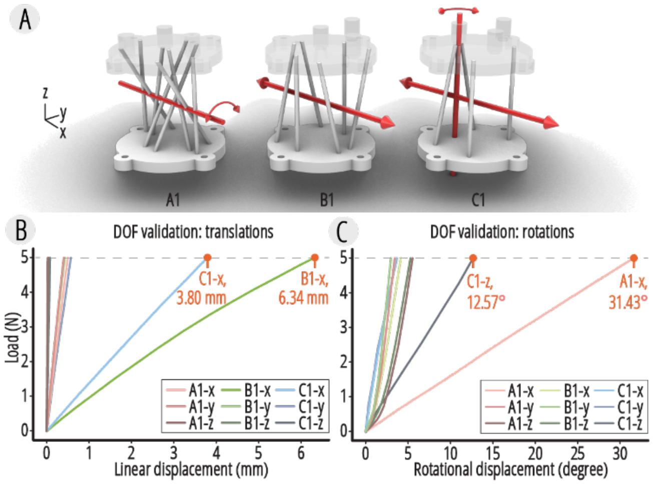


Figure 21: DOF design validation - (A) test samples and results for (B) translational and (C) rotational DOFs.

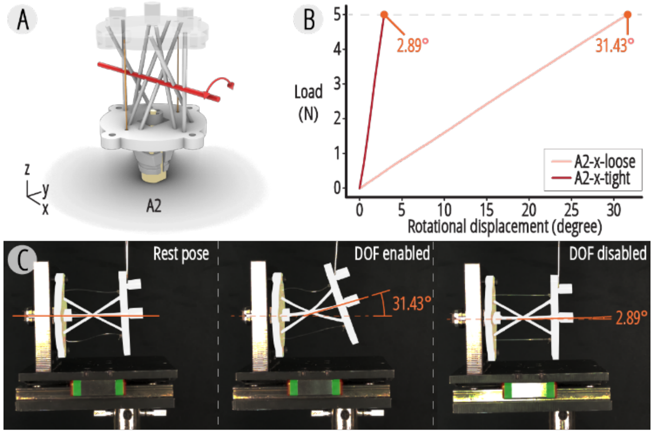


Figure 22: Rotational DOF reconfiguration test - (A) the test sample, (B) results, and (C) comparisons.

was 31.43 degrees when it was unlocked, but the mobility dropped to 2.89 degrees when the cables were tightened. B2 also showed a similar trend: under the same load, B2's translation along the x-axis dropped from 6.34 mm to 0.49 mm when it was locked (Figure 23B). These results indicate that the cable-driven reconfiguration was able to modulate the DOFs' stiffness by more than a magnitude, thus showing that the design tool and method produced truthful designs.

10 DISCUSSION AND LIMITATION

10.1 Kinematic Design Limitations

Although our evaluations and design examples show that the proposed framework can enable users to design a wide range of reconfigurable and multimodal kinematic mechanisms, certain types of kinematic joints are still unproducible. For instance, it is mechanically impossible to create a single CM joint that affords translation

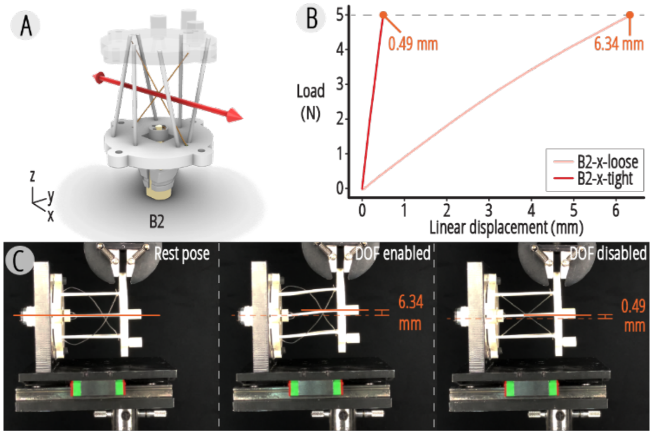


Figure 23: Translational DOF reconfiguration test - (A) the test sample, (B) results, and (C) comparisons.

along all three principal axes [6]. However, it is still possible to produce such a joint serially connecting two or more joints. I.e., a joint is valid as long as it has less than three translational DOFs, and the free end of a serial kinematic joint will have the combined DOFs of all joints.

10.2 Mechanical Design Considerations

On the other hand, compliant mechanisms also have certain kinematics limitations. Since the flexures enable DOFs by bending, it is impossible to create devices that allow continuous rotation. Translational CMs are also accompanied by parasitic displacements, meaning as the free end translates in one direction, it will also creep in a perpendicular direction. That said, we did not find this to be a major concern when developing the demonstrations, and users can add an orthogonal translational joint to compensate for this unwanted motion. Finally, structural fatigue has also been identified as a potential pitfall of CMs [21], but we did not observe this to be an issue even after months of repeated usage.

10.3 Device Kinetics Design

The framework focuses on designing the DOFs behaviors of a device, that is, kinematically speaking, the derivative of motions. We acknowledge that this is both an advantage and a limitation. Designing devices through their DOFs provides both force and dynamic responses to the user [43]. In this work, adding a reconfigurable and multimodal dimension navigates an even larger design space. However, the current state of the design tool and framework does not support other aspects of a device design, such as their kinetics - stiffness [41], motion range and trajectory [9, 20]. Designing these properties calls for an interactive and physically based simulator, and future work may consider adopting advanced or accelerated numerical simulations [43, 48] to provide these functions.

10.4 Scalability

Practically speaking, ReCompFig works best for a hand- to body-scale. While the design principles and algorithms are applicable across scales, the limitation comes from the material and components. When scaling the designs, there is a tradeoff between the joint's stiffness and weight-carrying capacity. Based on the Euler-Bernoulli beam theory, as a device scales up, the weight increases cubically while its load-carrying capacity increases quadratically, creating a need to thicken the flexures. Yet, a thicker flexure is structurally more vulnerable to deformations (scales linearly with the thickness). Therefore, users are advised to add more flexures instead while keeping the thickness constant or source for a stiffer material to circumvent the need to thicken the flexures. On the other hand, when scaling down the designs, the cables and motors were the bottlenecking factors as they are difficult to miniaturize and assemble. In this case, a novel method or material system is required.

11 FUTURE WORK

Designing kinematic devices is challenging due to its involvement of expert mechanical knowledge, and it is even harder to design these devices using compliant mechanisms as it involves nonlinear physics and topological kinematics. Yet, CMs also offer advantages unattainable by conventional mechanisms, such as their simplicity, ease of fabrication, precision, and scalability. We have only explored and utilized some of these properties in this work, and it would be exciting to see future design tools that are more powerful and enable/engage a wider range of users to adopt CMs in their design. For instance, can we produce design tools that automatically generate CM designs with different aesthetic and functional qualities? Can we develop intelligent fabrication software and tools that make CMs monolithically to reduce their fabrication complexity and assembly demand further, making them even more accessible to makers, researchers, and designers? Due to its complexity, CM design tools also make a good playground for human-computer collaboration studies.

Our demonstrations exemplified how ReCompFig devices can be leveraged to enable haptics in artificial reality. Yet, further evaluations are still needed to validate their applicability. We also speculate that combining the proposed framework with other interaction design methods may provide even more diverse, immersive, and augmentative experiences. On one hand, using the proposed framework and compliant mechanism design methods would allow future researchers to design shape-changing interfaces that have even more complex or dynamic tangible responses. On the other hand, incorporating shape- or stiffness-changing materials [28, 47] and other haptic modalities (e.g., texture [19], weight [26]) into kinematic devices may also provide more realistic and genuine sensations for artificial reality or produce perceptually and emotionally evocative interfaces.

12 CONCLUSION

This work introduces tension cables into compliant mechanisms to create multimodal and reconfigurable kinematic mechanisms and devices. We develop several design principles based on the screw theory of compliant mechanisms to govern and inform the design process. Technical contributions, including computational algorithms, design tools, fabrication methods, and validation, are also provided. In particular, the design tool assists users in designing two aspects of a reconfigurable CM device - its prescribed DOF configurability and the cables used for dynamically switching their DOF modes. The design tool provides procedural and open-ended guidance to assist users in creating mechanisms with the desired kinematic modes and sensing capabilities. Our evaluations also show that the design framework truthfully produces devices that have multimodal and reconfigurable kinematic behaviors. Design examples including material displays, haptic proxies, and a multimodal input device are also presented to showcase the mechanisms' application opportunities. With this vision, we publicize the tool at https://github.com/morphing-matter-lab/ReCompFig.

ACKNOWLEDGMENTS

This research is partially supported by Carnegie Mellon University's (CMU) Center for Machine Learning and Health and the National Science Foundation grant ID 052184116. We also thank the CMU's College of Engineering for supporting this work through the Moonshot program (4d6f6f6e73686f74). The authors also thank the reviewers' and colleagues' comments that helped improve the paper.

REFERENCES

[1] Jason Alexander, Andrés Lucero, and Sriram Subramanian. 2012. Tilt displays: designing display surfaces with multi-axis tilting and actuation. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12. https://doi.org/10.1145/2371574.2371600

[2] Siyeon Baik, Shinsuk Park, and Jaeyoung Park. 2020. Haptic Glove Using Tendon-Driven Soft Robotic Mechanism. Frontiers in bioengineering and biotechnology 8: 541105.

[3] Inrak Choi, Heather Culbertson, Mark R. Miller, Alex Olwal, and Sean Follmer. 2017. Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17), 119–130.

[4] Cathy Fang, Yang Zhang, Matthew Dworman, and Chris Harrison. 2020. Wireality: Enabling Complex Tangible Geometries in Virtual Reality with Worn Multi-String Haptics. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3313831.3376470

[5] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation. In Uist, 2501988–2502032.

[6] Jonathan B. Hopkins and Martin L. Culpepper. 2010. Synthesis of multi-degree of freedom, parallel flexure system concepts via Freedom and Constraint Topology (FACT) – Part I: Principles. Precision Engineering 34, 259–270. https://doi.org/10.1016/j.precisioneng.2009.06.008

[7] Larry L. Howell. 2013. Introduction to compliant mechanisms. In Handbook of Compliant Mechanisms. John Wiley & Sons Ltd, Oxford, UK, 1–13.

[8] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial Mechanisms. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16), 529–539.

[9] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. 2019. Understanding Metamaterial Mechanisms. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), 1–14.

[10] Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. 2017. Digital Mechanical Metamaterials. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 977–988.

[11] Hiroshi Ishii, Daniel Leithinger, Sean Follmer, Amit Zoran, Philipp Schoessler, and Jared Counts. 2015. TRANSFORM. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. https://doi.org/10.1145/2702613.2702969

[12] Sridhar Kota. 1999. Design of compliant mechanisms: applications to MEMS. Smart Structures and Materials 1999: Smart Electronics and MEMS. https://doi.org/10.1117/12.354294

[13] Robert Kovacs, Alexandra Ion, Pedro Lopes, Tim Oesterreich, Johannes Filter, Philipp Otto, Tobias Arndt, Nico Ring, Melvin Witte, Anton Synytsia, and Patrick

Baudisch. 2018. TrussFormer: 3D Printing Large Kinetic Structures. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18), 113–125.

[14] Robert Kovacs, Eyal Ofek, Mar Gonzalez Franco, Alexa Fay Siu, Sebastian Marwecki, Christian Holz, and Mike Sinclair. 2020. Haptic PIVOT: On-Demand Handhelds in VR. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20), 1046–1059.

[15] Andrey Krekhov, Katharina Emmerich, Philipp Bergmann, Sebastian Cmentowski, and Jens Krüger. 2017. Self-Transforming Controllers for Virtual Reality First Person Shooters. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play. Association for Computing Machinery, New York, NY, USA, 517–529.

[16] B. S. Lazarov, M. Schevenels, and O. Sigmund. 2011. Robust design of large-displacement compliant mechanisms. Mechanical sciences 2, 2: 175–182.

[17] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. 2014. Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14), 461–470.

[18] Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing Haptics to Walls & Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1471–1482.

[19] Qiuyu Lu, Jifei Ou, João Wilbert, André Haben, Haipeng Mi, and Hiroshi Ishii. 2019. milliMorph – Fluid-Driven Thin Film Shape-Change Materials for Interaction Design. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19), 663–672.

[20] Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. 2017. A computational design tool for compliant mechanisms. ACM transactions on graphics 36, 4: 1–12.

[21] E. G. Merriam, J. E. Jones, and S. P. Magleby. 2013. Monolithic 2 DOF fully compliant space pointing mechanism. Mechanical. Retrieved from https://ms.copernicus.org/articles/4/381/2013/

[22] Martin Murer, Bernhard Maurer, Hermann Huber, Ilhan Aslan, and Manfred Tscheligi. 2015. TorqueScreen: Actuated Flywheels for Ungrounded Kinaesthetic Feedback in Handheld Devices. In Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15), 161–164.

[23] Ken Nakagaki, Artem Dementyev, Sean Follmer, Joseph A. Paradiso, and Hiroshi Ishii. 2016. ChainFORM: A Linear Integrated Modular Hardware System for Shape Changing Interfaces. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16), 87–96.

[24] Ken Nakagaki, Daniel Fitzgerald, Zhiyao (john) Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. 2019. InFORCE. In Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction. https://doi.org/10.1145/3294109.3295621

[25] Ryosuke Nakayama, Ryo Suzuki, Satoshi Nakamaru, Ryuma Niiyama, Yoshihiro Kawahara, and Yasuaki Kakehi. 2019. MorphIO: Entirely Soft Sensing and Actuation Modules for Programming Shape Changes through Tangible Interaction. In Proceedings of the 2019 on Designing Interactive Systems Conference (DIS '19), 975–986.

[26] Ryuma Niiyama, Lining Yao, and Hiroshi Ishii. 2014. Weight and volume changing device with liquid metal transfer. In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction (TEI '14), 49–52.

[27] Jifei Ou, Zhao Ma, Jannik Peters, Sen Dai, Nikolaos Vlavianos, and Hiroshi Ishii. 2018. KinetiX - designing auxetic-inspired deformable material structures. Computers & graphics 75: 72–81.

[28] Jifei Ou, Lining Yao, Daniel Tauber, Jürgen Steimle, Ryuma Niiyama, and Hiroshi Ishii. 2014. jamSheets: thin interfaces with tunable stiffness enabled by layer jamming. In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction (TEI '14), 65–72.

[29] Simon Poppinga, Axel Körner, Renate Sachse, Larissa Born, Anna Westermeier, Linnea Hesse, Jan Knippers, Manfred Bischoff, Götz T. Gresser, and Thomas Speck. 2016. Compliant Mechanisms in Plants and Architecture. In Biomimetic Research for Architecture and Building Construction: Biological Design and Integrative Structures, Jan Knippers, Klaus G. Nickel and Thomas Speck (eds.). Springer International Publishing, Cham, 169–193.

[30] Simon Robinson, Céline Coutrix, Jennifer Pearson, Juan Rosso, Matheus Fernandes Torquato, Laurence Nigay, and Matt Jones. 2016. Emergeables: Deformable Displays for Continuous Eyes-Free Mobile Interaction. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/2858036.2858097

[31] Neung Ryu, Woojin Lee, Myung Jin Kim, and Andrea Bianchi. 2020. ElaStick: A Handheld Variable Stiffness Display for Rendering Dynamic Haptic Response of Flexible Object. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20), 1035–1045.

[32] Deepak Ranjan Sahoo, Timothy Neate, Yutaka Tokuda, Jennifer Pearson, Simon Robinson, Sriram Subramanian, and Matt Jones. 2018. Tangible Drops: A Visio-Tactile Display Using Actuated Liquid-Metal Droplets. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–14.

[33] Jeffrey A. Shank. 2008. Improvement and evaluation of three cable haptic interface. Ohio University. Retrieved December 20, 2021 from https://rave.ohiolink.edu/etdc/view?acc_num=ohiou1212701403

[34] Lucas A. Shaw, Frederick Sun, Carlos M. Portela, Rodolfo I. Barranco, Julia R. Greer, and Jonathan B. Hopkins. 2019. Computationally efficient design of directionally compliant metamaterials. Nature communications 10, 1: 291.

[35] Jotaro Shigeyama, Takeru Hashimoto, Shigeo Yoshida, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2019. Transcalibur: A Weight Shifting Virtual Reality Controller for 2D Shape Rendering based on Computational Perception Model. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–11.

[36] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational design of actuated deformable characters. ACM transactions on graphics 32, 4: 1–10.

[37] Frederick Sun and Jonathan B. Hopkins. 2017. Mobility and Constraint Analysis of Interconnected Hybrid Flexure Systems Via Screw Algebra and Graph Theory. Journal of mechanisms and robotics 9, 3. https://doi.org/10.1115/1.4035993

[38] Tao Sun, Shuofei Yang, and Binbin Lian. 2020. Finite and Instantaneous Screw Theory in Robotic Mechanism. Springer Nature.

[39] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19), 493–505.

[40] C. Swindells, A. Unden, and T. Sang. 2003. TorqueBAR: an ungrounded haptic feedback device. of the 5th international conference on .... Retrieved from https://dl.acm.org/doi/abs/10.1145/958432.958445

[41] Kyler A. Tolman, Ezekiel G. Merriam, and Larry L. Howell. 2016. Compliant constant-force linear-motion mechanism. Mechanism and Machine Theory 106: 68–79.

[42] Hsin-Ruey Tsai, Jun Rekimoto, and Bing-Yu Chen. 2019. ElasticVR: Providing Multilevel Continuously-Changing Resistive Force and Instant Impact Using Elasticity for VR. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–10.

[43] Omer Anil Turkkan, Venkatasubramanian Kalpathy Venkiteswaran, and Hai-Jun Su. 2018. Rapid conceptual design and analysis of spatial flexure mechanisms. Mechanism and Machine Theory 121: 650–668.

[44] Venkatasubramanian Kalpathy Venkiteswaran, Omer Anil Turkkan, and Hai-Jun Su. 2016. Compliant Mechanism Design Through Topology Optimization Using Pseudo-Rigid-Body Models. Volume 5A: 40th Mechanisms and Robotics Conference. https://doi.org/10.1115/detc2016-59946

[45] Luke Vink, Viirj Kan, Ken Nakagaki, Daniel Leithinger, Sean Follmer, Philipp Schoessler, Amit Zoran, and Hiroshi Ishii. 2015. TRANSFORM as Adaptive and Dynamic Furniture. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15), 183.

[46] Guanyun Wang, Tingyu Cheng, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao. 2018. Printed Paper Actuator: A Low-cost Reversible Actuation and Sensing Method for Shape Changing Interfaces. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), 1–12.

[47] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. 2019. A-line: 4D Printing Morphing Linear Composite Structures. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–12.

[48] Humphrey Yang, Kuanren Qian, Haolin Liu, Yuxuan Yu, Jianzhe Gu, Matthew McGehee, Yongjie Jessica Zhang, and Lining Yao. 2020. SimuLearn: Fast and Accurate Simulator to Support Morphing Materials Design and Workflows. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20), 71–84.

[49] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13), 13–22.

[50] Lining Yao, Jifei Ou, Chin-Yi Cheng, Helene Steiner, Wen Wang, Guanyun Wang, and Hiroshi Ishii. 2015. BioLogic: natto cells as nanoactuators for shape changing interfaces. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 1–10.

[51] Andre Zenner and Antonio Kruger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. IEEE transactions on visualization and computer graphics 23, 4: 1285–1294.