

Conformable Mechanisms on Freeform Surfaces

Siqi Li^a, Haoyu Tang^a, Peng Song^a, Bailin Deng^b, Jianmin Zheng^c

^aSingapore University of Technology and Design, Singapore

^bCardiff University, United Kingdom

^cNanyang Technological University, Singapore

ARTICLE INFO

Article history:

Received July 22, 2025

Keywords: Conformable mechanism

Path generation

Motion generation

Freeform surface

Computational design

ABSTRACT

This paper introduces a new class of linkage mechanisms called *surface-conformable mechanisms* or simply *conformable mechanisms*. A conformable mechanism conforms to a freeform surface in one of its configurations, in which the mechanism's joints and links are exactly on the surface. Conformable mechanisms can be stowed compactly when not in use and accomplish complex motion transfer tasks when deployed. This paper aims to model and design conformable mechanisms for 3D path and motion generation. To achieve this goal, we enumerate topologies of conformable mechanisms, and model their geometry in the parameterization space of a freeform surface for surface conformity. To ensure a working and fabricable mechanism, we propose an efficient approach to processing the freeform surface by first removing a portion of the surface that collides with the moving links and joints and then removing disconnected patches and fragile features from the surface. Taking the modeling and processing as a foundation, we propose an optimization-based approach to designing a conformable mechanism for generating a target 3D path/motion, while preserving the mechanism's appearance in the stowed state. We demonstrate the effectiveness of our approach by designing conformable mechanisms that conform to various freeform surfaces, evaluating their kinematic performance in 3D path and motion generation, validating their functionality with a 3D printed prototype, and showing three applications of these mechanisms.

© 2025 Elsevier B.V. All rights reserved.

1. Introduction

This paper introduces a new class of linkage mechanisms whose joints and links conform to a freeform surface in one of its configurations. We name such mechanisms *surface-conformable mechanisms*, or simply *conformable mechanisms*; see Figure 1 for an example. Conformable mechanisms can be stowed compactly by conforming to a freeform surface and deployed from that surface to accomplish complex motion transfer tasks. Due to this advantage, conformable mechanisms are suitable to be used for devices that should be compact and functional, where the freeform surface can be the device's enclosure surface. Typical examples of such devices include mechanical toys for recreation, medical instruments for minimally invasive surgery, and spacecrafts that are launched in a compact form.

A spatial linkage is a conventional mechanism where links connected by mechanical joints are able to move in the 3D space [1]. A conformable mechanism can be considered as a spatial linkage where all the joints and links are exactly on a freeform surface in one of its configurations, making the spatial linkage compact in that configuration. Our conformable mechanism is inspired by an emerging mechanism called developable mechanism, which is a linkage mechanism that conforms to a developable surface and consists of revolute joints only [2]. Compared with developable mechanisms, our conformable mechanisms can be used for a wider variety of applications due to two advantages. First, conformable mechanisms can be used for devices with more general enclosure shapes since they conform to freeform surfaces, which are much more

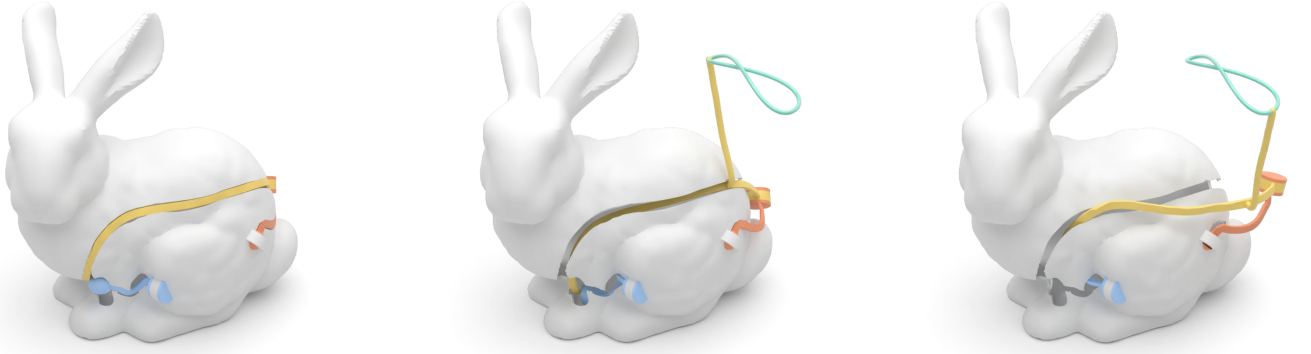


Fig. 1. We present a computational approach for modeling and designing a new class of linkage mechanisms called *conformable mechanism*. A conformable mechanism conforms to a freeform surface in its stowed state (left) and deploys from that surface to accomplish a motion transfer task (middle and right); see the 3D path (in green color) generated by the mechanism. Thanks to the surface conformity, our conformable mechanism is compact in its stowed state.

general than developable surfaces, to which developable mechanisms conform. Second, conformable mechanisms are able to generate a wider variety of paths/motions since they allow using a combination of different kinds of mechanical joints (e.g., spherical joint, revolute joint) while developable mechanisms only allow using revolute joints according to their definition.

In this paper, we aim to model and design conformable mechanisms for generating a target 3D path/motion while conforming to a specified freeform surface. This problem is non-trivial due to three reasons. First, geometric design of linkage mechanisms for 3D path/motion generation is a fundamental and challenging problem [1]. Second, the surface conformity requirement enforces a hard constraint on the position of each joint and the shape of each link, making the problem harder. Lastly, to ensure a working mechanism, part of the freeform surface has to be removed to avoid collision between the static surface and the mechanism in motion. However, a simple cut on the surface may make the mechanism unfabricable and negatively impact the mechanism's appearance.

We present techniques to address the above challenges. Our main contributions are as follows.

- We propose a unified approach to modeling the topology, geometry, and kinematics of conformable mechanisms. To ensure a working and fabricable mechanism, we propose an efficient approach to processing the freeform surface of the mechanism.
- We propose an optimization-based approach to designing a conformable mechanism for generating a target 3D path/motion, while preserving the mechanism's appearance in the stowed state.

We show that conformable mechanisms are able to conform to freeform surfaces with a variety of shapes, evaluate their kinematic performance in 3D path/motion generation, and validate their functionality with a 3D printed prototype. We demonstrate the usefulness of conformable mechanisms with three applications, i.e., mechanical toy, ceiling-mounted manipulator, and mobile robot.

2. Related Work

Mechanism Design. Mechanism design involves modeling a mechanism to meet a set of kinematic requirements. Two fundamental goals of mechanism design are *path generation* and *motion generation* (also called *rigid body guidance*). Path generation aims to design a mechanism to make a point move along a prescribed trajectory. Researchers in the graphics community have proposed computational approaches to address the path generation problem for designing mechanical characters [3, 4], drawing devices [5], and walking machines [6]. Recently, researchers proposed modeling unconventional mechanisms for addressing the path generation problem, including compliant mechanisms [7, 8], 3D cam-follower mechanisms [9], and 3D cam-linkage mechanisms [10]. Rather than focusing on moving a point, motion generation aims to design a mechanism for guiding a rigid body through a series of specified positions and orientations. Researchers have addressed this problem to design various fabricable mechanical automata, including mechanical toys [11], mechanical figures that mimic human motions [12], planar mechanical characters [13], wind-up toys [14], and kinetic wire characters [15]. Recently, Chen et al. [16] modeled a multi-point conjugation mechanism for exactly generating a prescribed 3D motion, which only consists of two moving parts.

In this paper, our goal is to model a conformable mechanism for 3D path/motion generation. The unique characteristic of our conformable mechanism is that it conforms to a freeform surface in one of its configurations, making it compact in that configuration.

Linkage Mechanism. Conventionally, linkage mechanisms can be classified into planar linkages, spherical linkages, and spatial linkages [1]. Planar linkages are able to generate 2D paths/motions in parallel planes [17, 18, 19, 20] while spherical linkages are able to generate 2D paths/motions on concentric spheres [21, 22]. Different from planar and spherical linkages, spatial linkages are able to generate 3D paths/motions by allowing posing different kinds of mechanical joints in the 3D space. To generate a 3D path, the most typical way is to use

1-DOF spatial linkages driven by a single actuator. These spatial linkages can approximate a prescribed 3D path [23, 24, 25] or generate a path that exactly passes through a few prescribed 3D points (e.g., 9 precision points for RCCC linkages [26]). 1-DOF spatial linkages also have been designed to guide a rigid body through a few prescribed 3D poses [27].

Our conformable mechanism is a spatial linkage that conforms to a freeform surface. The surface conformity feature enables new functionalities that cannot be achieved by conventional spatial linkages. For example, conformable mechanism can be stowed compactly in a device's enclosure surface and deployed from that surface.

Developable Mechanism. A developable mechanism is a linkage (with revolute joints only) that conforms to a developable surface at one configuration in its motion by aligning its revolute joint axes with the ruling lines of the developable surface [2]. Besides joints, the links of a developable mechanism must also conform to the developable surface [28]. Developable mechanisms have been modeled by mapping planar, spherical, or spatial linkages onto developable surfaces [29, 30]. In particular, planar linkages can be mapped to cylindrical surfaces [28], spherical linkages can be mapped to conical surfaces [31], and spatial linkages can be mapped to tangent developable surfaces [2]. Developable mechanisms can also be classified according to the motion of a developable mechanism relative to the developable surface, i.e., intramobile, extramobile, and transmobile developable mechanisms [28, 32]. Recently, there was an interest in designing developable mechanisms for 2D path/motion generation. Sheffield et al. [33] proposed a numerical approach to designing cylindrical developable mechanisms that can perform linear motion, by embedding existing linear-motion linkages into a cylindrical surface. Vennard et al. [34] proposed two approaches (one analytical and one graphical) to designing cylindrical developable mechanisms for 2D path/motion generation.

Conformable mechanisms generalize developable mechanisms in two aspects: conforming to more general surfaces and supporting different combinations of joint types. Due to this reason, conformable mechanisms have a larger number of design parameters (i.e., joint type and joint orientation) than developable mechanisms. Thanks to the generalization, conformable mechanisms can conform to surfaces with a wide variety of shapes (see Figure 8) and generate a wide variety of paths/motions (see Figures 10 and 11).

3. Problem Definition

We define a conformable mechanism as a mechanism that meets the following conditions:

1. the mechanism consists of links connected by joints;
2. the mechanism conforms to a freeform surface when both the joints and links are modeled with zero thickness;
3. the mechanism has mobility; and
4. the mechanism's movement neither causes a collision with the surface nor requires the surface to deform.

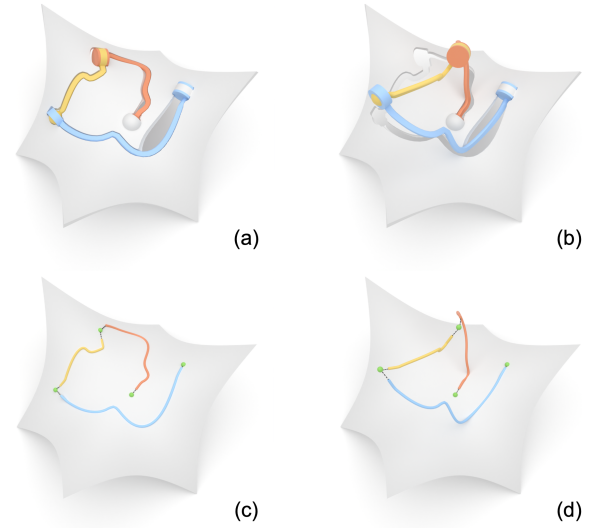


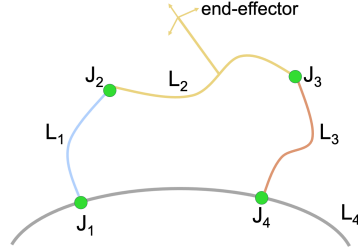
Fig. 2. A conformable mechanism, consisting of four joints and four links, (a) conforms to a freeform surface and (b) deploys from that surface to generate a 3D motion. Note that the supporting link (in gray color) is the processed and thickened reference surface. (c&d) The mechanism's joints (in green color) and links are modeled with zero thickness to show its surface conformity and mobility relative to the surface, respectively.

The freeform surface that a conformable mechanism conforms to is called the mechanism's *reference surface*. The reference surface is typically the enclosure surface of a device that makes use of a conformable mechanism. In this paper, we focus on conformable mechanisms driven by a *single actuator*, i.e., conformable mechanisms with one degree of freedom (DOF); see Figure 2 for an example. However, the definitions and techniques presented in this paper can be generalized to multi-DOF conformable mechanisms.

Problem formulation. Our goal is to model and design a conformable mechanism that generates a target 3D path/motion while conforming to a reference surface, driven by a single actuator. A target 3D path is represented as a sequence of 3D points while a target 3D motion is represented as a sequence of 3D poses (i.e., positions and orientations). The reference surface is prescribed by users and the target 3D path/motion is specified relative to the reference surface. We assume that the single actuator performs periodic motion, which can be either rotational motion or oscillating motion. Given the target path/motion and the reference surface, we model and design a conformable mechanism satisfying the following requirements:

1. *Working mechanism.* The conformable mechanism should be a working mechanism, where the motion can be *smoothly* transferred from the driving link to the end-effector without *collisions* and *singularities*.
2. *Surface conformity.* The conformable mechanism should conform to the reference surface at one of its configurations, forming a stowed state.
3. *Path/motion generation.* The conformable mechanism should be able to generate the target 3D path/motion, meaning that its end-effector should (exactly or approximately) pass through the target set of 3D points/poses.

Fig. 3. A 2D illustration of a 3D conformable mechanism with $K = 4$ joints, where the supporting link L_4 is the reference surface after processing and thickening.



4. *Mechanism mobility.* The conformable mechanism's mobility relative to the reference surface should support its functionality; e.g., a mechanism that performs manipulation tasks should always be out of its reference surface.
5. *Fabricable mechanism.* The conformable mechanism should be fabricable (e.g., with 3D printing), where fragile features should be avoided on each mechanical part, especially for the supporting part generated via processing and thickening the reference surface.
6. *Mechanism appearance.* A certain portion of the reference surface has to be removed to avoid collision between the surface and the mechanism in motion. Thus, the conformable mechanism may show gaps in the stowed state, compared with the original reference surface. To preserve the mechanism's appearance, the total area of these gaps should be as small as possible.

Overview of our approach. We first introduce an approach to model conformable mechanisms, including enumerating the mechanisms' topologies, parameterizing their geometry, computing their forward kinematics, and identifying their mobility relative to the reference surface; see Section 4. Next, we propose a method to process the geometry of the reference surface, ensuring collision-free motion of the mechanism and fabricability of the processed surface; see Section 5. Lastly, we propose an optimization-based approach to designing a conformable mechanism satisfying the above requirements; see Section 6.

4. Modeling Conformable Mechanisms

In this section, we introduce an approach to modeling topology, geometry, and kinematics of conformable mechanisms.

4.1. Topology Modeling

In this paper, we only consider simple topologies of conformable mechanisms. We assume that a conformable mechanism consists of several joints that form a closed chain, where a link connects two consecutive joints; see Figure 3. Under this assumption, the topology of a conformable mechanism is defined by the permutation of joints in the linkage mechanism. Moreover, we only consider mechanical joints that allow pure rotational motion since these joints are easy to conform to a freeform surface due to their small sizes. Table 1 presents three types of such joints, R, U, and S, which allow 1-DOF, 2-DOF, and 3-DOF rotational motion, respectively. In particular, the driving joint of the mechanism has to be an R joint since the actuator always performs 1-DOF rotational motion.

Table 1. Three types of joints used in our conformable mechanism, as well as their allowed motions. Each joint is visualized in its default configuration, where the socket part is rendered in yellow and the pin part is rendered in orange.

	Joint name	Abbrev.	Motion	s_k	Motion Range
	Revolute joint	R	1-DOF Rotation	α	$\alpha \in [-180^\circ, 180^\circ]$
	Universal joint	U	2-DOF Rotation	$[\alpha, \beta]$	$\alpha \in [-180^\circ, 180^\circ]$ $\beta \in [-45^\circ, 45^\circ]$
	Spherical joint	S	3-DOF Rotation	$[\alpha, \beta, \gamma]$	$\alpha \in [-180^\circ, 180^\circ]$ $\beta \in [-45^\circ, 45^\circ]$ $\gamma \in [-45^\circ, 45^\circ]$

We denote each joint in a conformable mechanism by J_k and its number of DOFs by f_k , where the first joint J_1 is the driving joint. The numbers of DOFs $\{f_k\}$ of all the joints $\{J_k\}$ should satisfy the mobility formula [35]:

$$6(N - 1 - K) + \sum_{k=1}^K f_k = M, \quad (1)$$

where N is the number of links, K is the number of joints, and $M = 1$ is the mechanism's DOF. Since $N = K$ for a closed-chain linkage mechanism, the mobility formula is simplified as:

$$\sum_{k=1}^K f_k = 7, \quad (2)$$

In practice, a closed-chain linkage with 4 or 5 joints are most widely used. Hence, we focus on conformable mechanisms with 4 or 5 joints (i.e., $K \in \{4, 5\}$). Note that Equation 2 is applicable to conformable mechanisms with more joints.

Based on Equation 2, we are able to enumerate all the valid topologies of our conformable mechanism. For a conformable mechanism with $K = 4$ joints, we have 7 topologies, i.e., RRUS, RRSU, RURS, RUUU, RUSR, RSRU, RSUR. For a conformable mechanism with $K = 5$ joints, we have 10 topologies, i.e., RRRRS, RRRUU, RRRSR, RRURU, RRUUR, RRSRR, RURRU, RURUR, RUURR, RSRRR.

4.2. Geometry Modeling

We refer to the configuration where a conformable mechanism conforms to its reference surface as the *conforming configuration*. We propose to model the geometry of a conformable mechanism in its conforming configuration, aiming to satisfy the surface conformity requirement. We choose to parameterize the mechanism's geometry in the reference surface's parameterization space rather than in the 3D Euclidean space since it involves fewer parameters.

We first introduce our parameterization of the reference surface, followed by our approach to modeling each component of the mechanism.

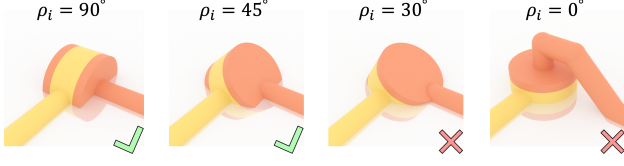


Fig. 4. A revolute joint with four different orientations relative to the same reference surface, among which two orientations (right) lead to violation of the surface conformity requirement by one associated link (in orange).

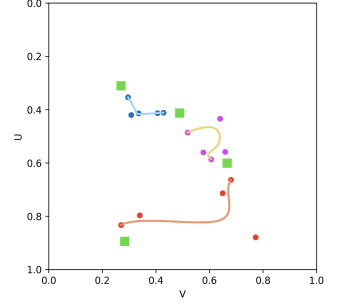
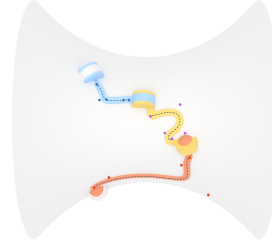


Fig. 5. (Left) A conformable mechanism in its stowed state. (Right) Modeling the centerline of each moving link as a 2D B-Spline curve in the UV space of the reference surface, where the joint centers are represented as green squares.

Reference surface parameterization. We assume that the reference surface S of a conformable mechanism is parameterized by a bijective, smooth function $\mathbf{r} : D \rightarrow \mathbb{R}^3$, where $D \subset \mathbb{R}^2$ is a subset of the plane and chosen as a square $[0, 1] \times [0, 1]$. The reference surface S is the image of \mathbf{r} , so that $S = \{\mathbf{r}(u, v) : (u, v) \in D\}$. According to this assumption, the reference surface should be homeomorphic to a disk. In case the reference surface's topology is more complex than a disk, we introduce cuts on the surface to flatten it. The bijective and smooth parameterization ensures that the mechanism geometry modeled in the parameter space can be uniquely and smoothly mapped onto the reference surface. In our experiments, we choose discrete conformal maps [36] for the reference surface parameterization.

A conformable mechanism is a spatial linkage whose joints and links conform to a freeform surface. Among all the joints, the driving joint J_1 and the last joint J_K are always fixed on the reference surface while the other joints are able to move in the 3D space; see again Figure 3. Among all the links, the supporting link L_K connects joint J_1 and joint J_K via the reference surface while the other links are able to move in the 3D space. To output the generated path/motion, an end effector is attached with one moving link of the mechanism. The details of modeling the geometry for these components in the conforming configuration are elaborated below.

i) Joints. A joint consists of two parts, a socket and a pin, and allows relative motion between the two parts. Table 1 shows three different types of joints used in a conformable mechanism, i.e., R, U, and S joints, as well as their allowed relative motion s_k . Note that the U joint is a modified S joint that removes 1-DOF rotation by inserting two pins in the S joint. The geometry of each joint defines the range of relative motion between its two parts. Due to the fabrication constraint, both U and S joints can support $[-180^\circ, 180^\circ]$ rotation around the spinning axis and $[-45^\circ, 45^\circ]$ rotation for each of the other rotation axes.

In the conforming configuration, the center of each joint J_k should be exactly on the reference surface, represented as $\mathbf{r}(u_k, v_k)$, where $(u_k, v_k) \in [0, 1] \times [0, 1]$ is the UV coordinate of the joint center. In the conforming configuration, the rotation axis of each joint J_k is defined by (θ_k, ϕ_k) in the world coordinate system (i.e., the coordinate system of the reference surface in the Euclidean space), where $\theta_k \in [0, 2\pi]$ is the azimuthal angle and $\phi_k \in [0, \pi]$ is the polar angle. In particular, for joint type U, its two pins are always perpendicular to the rotation axis. Hence, we use one more angle to define the orientation of the two pins relative to the rotation axis. For simplicity, we denote orientation parameters for each joint by Φ_k . Besides, there is a constraint on the joint orientation. Denote the angle between a joint J_k 's spinning axis and the reference surface

normal by ρ_k in the conformed configuration. We require that $\rho_k \in [\pi/4, 3\pi/4]$ such that a link connecting joint J_k is able to conform to the reference surface; see Figure 4 for examples.

We connect multiple joints following a pattern to form a conformable mechanism; see Figure 2 for an example. In particular, the driving joint J_1 is the R joint and we fix its socket part with the reference surface. We connect a joint J_k 's pin part to the next joint J_{k+1} 's socket part, except for the last joint J_K . Similar to the driving joint J_1 , we fix the last joint J_K 's socket part with the reference surface and connect its pin part with the previous joint J_{K-1} 's pin part.

ii) Moving links. A moving link L_i , $i \in [1, K-1]$ connects two consecutive joints J_i and J_{i+1} , among which the moving link L_1 is the driving link; see again Figure 3. We model each moving link L_i as a curved bar with a rectangular cross-section. Furthermore, we model the centerline of each moving link L_i as a 2D B-spline with m control points in the surface's UV space, where m is set to 5 in our experiments; see Figure 5. This ensures that the link's 3D centerline is exactly on the reference surface in the conforming configuration. We denote m control points of a moving link L_i 's centerline by $\mathbf{c}_{i,j}$, $j \in [1, m]$. Each of the two endpoints of a moving link L_i 's centerline should connect to a corresponding joint. To satisfy surface conformity in the conformed configuration, the endpoint $\mathbf{c}_{i,1}$ ($\mathbf{c}_{i,m}$) should locate at the intersection between the joint J_i (joint J_{i+1}) and the reference surface. In our experiments, the endpoint $\mathbf{c}_{i,1}$ ($\mathbf{c}_{i,m}$) are selected as a point on the intersection that is nearest to its next (previous) control point $\mathbf{c}_{i,2}$ ($\mathbf{c}_{i,m-1}$).

iii) Supporting link. The supporting link L_K connects two joints J_1 and J_K fixed on the reference surface. We model the supporting link L_K by processing and thickening the reference surface. To avoid collision between each moving link/joint and the supporting link, we have to modify the geometry of the supporting link L_K (i.e., remove some regions of the thickened reference surface) while ensuring that it is still fabricable; see Section 5 for details.

iv) End effector The end effector is modeled as a small ball (a local frame) for 3D path (motion) generation. The end effector should be attached to one moving link L_i , $i \in [1, K-1]$, in order to output the generated path/motion. Since the first link L_1 can only output 1-DOF rotational motion (i.e., the driving motion), it cannot be used to put the end effector. Without loss

of generality, we always attach the end effector to the link L_2 ; see again Figure 3. We use a bar to connect the end effector to the link L_2 , whose centerline is modeled as a 3D B-spline. The 3D B-spline has five control points, among which the first control point is on the link L_2 and the last control point is on the end-effector. We assume that the end-effector is disassemblable and does not need to conform to the reference surface.

4.3. Kinematic Modeling

We now explain how to model the forward kinematics of a conformable mechanism with known topology and geometry and how to identify the mobility type of a conformable mechanism relative to its reference surface.

Forward kinematics. Forward kinematics of a mechanism determines the mechanism's state for a given actuation parameter. The state \mathbf{s} of a conformable mechanism is defined through its time-varying joint angles, i.e., $\mathbf{s} = [\mathbf{s}_1(t), \mathbf{s}_2(t), \dots, \mathbf{s}_K(t)]$; see Table 1 for details. The actuation parameter of our conformable mechanism is the angle of the driving joint J_1 , i.e., \mathbf{s}_1 . The forward kinematics of the mechanism can be expressed as the following motion transfer function:

$$[\mathbf{s}_2, \dots, \mathbf{s}_K] = \mathbf{f}_{\text{linkage}}(\mathbf{s}_1) \quad (3)$$

Assuming the linkage's initial state $\bar{\mathbf{s}} = \mathbf{s}(0)$ is known and valid, the constraint on the state \mathbf{s} at time t enforced by the single loop in our conformable mechanism can be formulated as:

$$\mathbf{C}_L(\mathbf{s}_1, \dots, \mathbf{s}_K) = \mathbf{0} \quad (4)$$

where \mathbf{C}_L is a column vector with 12 elements. We compute the constraint \mathbf{C}_L following the approach in [10]. Given the known actuation parameter \mathbf{s}_1 , we can then solve for the unknown passive joint angles $[\mathbf{s}_2, \dots, \mathbf{s}_K]$ of the mechanism by minimizing a non-linear least squares problem:

$$\min_{\mathbf{s}} \mathbf{C}_L^T \mathbf{C}_L \quad (5)$$

Once we solve the forward kinematics $\mathbf{f}_{\text{linkage}}(\cdot)$ for the joint angles, the end-effector's pose $\mathbf{M}_e(t)$ can be calculated using:

$$\mathbf{M}_e(t) = \mathbf{M}_{L_2}(t) \bar{\mathbf{M}}_e, \quad t \in [0, T] \quad (6)$$

where $\mathbf{M}_{L_2}(t)$ is a matrix that represents the transformation of link L_2 at time t , and $\bar{\mathbf{M}}_e$ is a constant matrix that represents the end effector's pose in link L_2 's local coordinate system. When the driving joint J_1 moves for a whole period T , $\mathbf{M}_e(t)$ is the motion generated by the conformable mechanism.

We analyze singularities of the mechanism's forward kinematics following the approach in [37]. Differentiating Equation 4 with respect to time t leads to the relationship between the input and output speeds as follows:

$$\mathbf{A} [\dot{\mathbf{s}}_1] + \mathbf{B} [\dot{\mathbf{s}}_2, \dots, \dot{\mathbf{s}}_K] = \mathbf{0}$$

where $\mathbf{A} = \frac{\partial \mathbf{C}_L}{\partial \mathbf{s}_1}$ and $\mathbf{B} = \frac{\partial \mathbf{C}_L}{\partial [\mathbf{s}_2, \dots, \mathbf{s}_K]}$ are both Jacobian matrices. The minimal singular value $\sigma_{\min}(\mathbf{B})$ represents the distance from the mechanism state $\mathbf{s}(t)$ to the singularities.

Mobility type identification. We classify the mobility of a conformable mechanisms into three types according to its motion relative to the reference surface following [28]:

1. *intramobile* conformable mechanism that completely enters the interior of the reference surface when it moves from the conforming configuration;
2. *extramobile* conformable mechanism that completely enters the exterior of the reference surface when it moves from the conforming configuration; and
3. *transmobile* conformable mechanism that simultaneously enters both the interior and exterior of the reference surface when it moves from the conforming configuration.

Note that when the reference surface S is an open surface, its interior is defined by the volume between the surface itself and the surface's projection onto the ground.

To identify the mobility type of a given conformable mechanism M , we first compute the mechanism M 's forward kinematics for a whole period T . Denote the surface swept by the centerline of a moving link L_i during that motion by C_{L_i} , and the curve swept by the center point of a moving joint J_k during that motion by C_{J_k} . Denote the whole set of swept surfaces and swept curves by $W = \{C_{L_i}, C_{J_k}\}$, where $i \in [1, K-1]$, $k \in [2, K-1]$. According to the above definition, we identify the mobility type of the mechanism M using:

$$\text{Mobility}(M, S) = \begin{cases} \text{Intramobile} & \text{if } \forall \mathbf{x} \in W, O_S(\mathbf{x}) \in \{-1, 0\}, \\ \text{Extramobile} & \text{if } \forall \mathbf{x} \in W, O_S(\mathbf{x}) \in \{1, 0\}, \\ \text{Transmobile} & \text{otherwise.} \end{cases} \quad (7)$$

where $O_S(\mathbf{x})$ is a sign function for the reference surface S , and a value of -1, 0, and 1 for $O_S(\mathbf{x})$ means that a 3D point \mathbf{x} is in, on, and out of the surface S , respectively.

5. Processing Reference Surfaces

To make a conformable mechanism working and fabricable, we need to process the reference surface S to form the supporting link L_K . This surface processing has to be computationally efficient since it will be repeatedly executed when designing a conformable mechanism in Section 6. We present an approach for reference surface processing, consisting of four steps:

(i) *Cutting the reference surface.* During the motion of a conformable mechanism, each moving link and moving joint may collide with the reference surface. These collisions have to be resolved via cutting the reference surface so that the mechanism can work in practice. To perform the surface cutting, one straightforward approach is to compute a swept volume for each moving link/joint and then perform a CSG difference operation with the reference surface. However, this approach is computationally expensive, making it unsuitable for our problem. We propose an efficient though less precise approach to cut the reference surface. First, we remesh the reference surface with a dense set of triangles (around 10,000 triangles in our experiments), and construct a bounding volume hierarchy

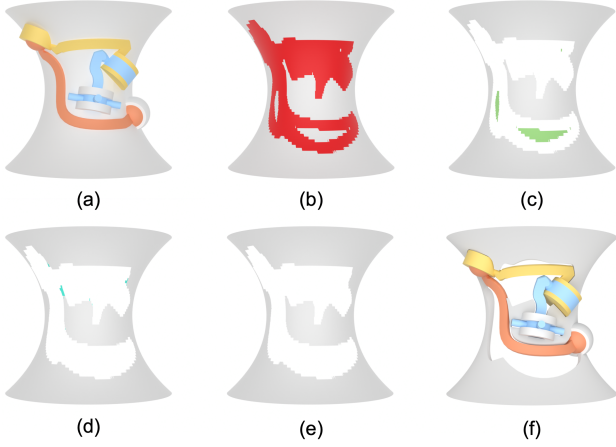


Fig. 6. Processing a reference surface. (a) A conformable mechanism and its original reference surface. (b) Detect triangles (in red) on the surface that collide with the mechanism during its motion. (c) Remove colliding triangles and identify disconnected surface patches (in green). (d) Remove disconnected patches and detect fragile features (in blue) in the remaining surface. (e) Remove fragile features from the remaining surface. (f) The conformable mechanism with the processed reference surface.

using axis-aligned bounding boxes (AABB) for the surface triangles. Collision detection between each moving link/joint's bounding volume and the reference surface's bounding volume hierarchy is performed at each sampled time $t \in [0, T]$. The regions on the reference surface that intersect with any moving link/joint are identified and removed; see Figure 6(a&b). In practice, the complex motion of a conformable mechanism may lead to a scattered cutting pattern, resulting in disconnected surface patches and/or fragile surface features; see Figure 6(c).

(ii) *Removing disconnected surface patches.* After the surface is cut, we check if the remaining surface is a piece of connected geometry by running a breadth-first search (BFS) over the triangle mesh of the surface, starting from a triangle intersecting with the driving joint J_1 's socket part. The remaining surface is connected if all its triangles are visited by the BFS. Otherwise, triangles unvisited by the BFS are considered as disconnected surface patches and are removed from the remaining surface; see Figure 6(c&d). Although our approach guarantees that the driving joint J_1 is always on the remaining surface, the other fixed joint J_K may not be on the remaining surface. In this case, we consider that the surface processing fails and terminate it.

(iii) *Removing fragile surface features.* The previous step ensures that the remaining surface is a piece of connected geometry. Yet, it may still contain fragile features that are unsuitable for fabrication. Hence, we have to identify these fragile features and remove them from the remaining surface; see Figure 6(d&e). The problem of identifying fragile features in general 3D shapes has been addressed by [38]. Although this method is powerful, we did not choose it due to its low efficiency (i.e., a few minutes to analyze a single 3D shape). Instead, we solve our problem by modifying the approach in [39] that efficiently identifies thin areas in 3D shapes represented as voxelizations by using top-hat transforms:

$$T_{\text{top-hat}}(\Omega) = \Omega \setminus T_l(D(T_h(D(\bar{\Omega}), \tau)), \tau) \quad (8)$$

where Ω is the input model, $\bar{\Omega} = \mathbb{R}^3 \setminus \Omega$ represents the boundary of the 3D model, D is a distance transform with respect to the Euclidean distance, T_h and T_l are the sets obtained by thresholding at level τ and by keeping only values higher and lower than τ , respectively. The regions identified using the top-hat transform $T_{\text{top-hat}}(\Omega)$ consists of fragile features and boundary regions. Hence, boundary regions are further subtracted from the identified regions.

The modification that we make is to redefine terms and operations in Equation 8 such that it is able to detect fragile features in 3D shapes represented as surface meshes. Specifically, Ω is the remaining surface represented as a triangle mesh, $\bar{\Omega}$ represents boundary of the 3D surface, D is a distance transform with respect to the *geodesic distance* over the triangle mesh, and T_h and T_l have the same meaning as above. To identify fragile features on the remaining surface, we define the local thickness t of a point p on a surface S by:

$$t(p) = 2\max(r|p \in B(x, r) \subset S, x \in S), \quad (9)$$

where $B(x, r)$ is a geodesic circle on the surface S centered at x with radius r . Fragile features are defined as the set of points on the surface S whose local thickness is smaller than a threshold τ . In our implementation, we compute the geodesic distance on the surface mesh using the method in [40], and set the threshold τ as 0.5cm.

After identifying the fragile features, we remove them from the remaining surface. However, this removal operation may introduce new disconnected patches on the surface. In this case, we repeat the previous step to eliminate them.

(iv) *Evaluating mechanism appearance.* After completing the above steps, we obtain a fabricable surface with a certain portion removed, denoted by S' . As aforementioned, the mechanism appearance is evaluated based on gaps of the mechanism in the stowed state using

$$E_{\text{gap}} = \text{Area}(S \setminus (G_M \cup S')), \quad (10)$$

where G_M is the intersections between the mechanism in the stowed state and the original reference surface S . A small E_{gap} means a small total area of gaps of the mechanism in the stowed state, which is preferred.

Remark. Our surface processing algorithm is computationally efficient. The time complexity of step (i) is $O(LK \log(n))$, where L is the number of sampled time stamps in the motion period T , K is the number of joints, and n is the number of triangles in the reference surface mesh. The time complexity of step (ii) is $O(n)$ due to the BFS over the surface mesh. The time complexity of step (iii) is $O(n \log(n))$ due to computing the distance transform over the surface mesh [40]. In our experiments, it typically takes 0.04s to process a surface mesh with around 10,000 triangles. Due to the efficiency, our surface processing algorithm is used for quantifying a mechanism design candidate based on Equation 10. Once a design is obtained, we process the reference surface using the same algorithm except cutting the surface in step (i) using the swept volume based method, aiming to obtain a smoother surface boundary; compare Figure 6(e&f).

6. Designing Conformable Mechanisms

We now present our approach to designing a conformable mechanism satisfying the requirements listed in Section 3. The user inputs include a reference surface S , a target 3D path/motion represented by a sequence of points/poses $\{\mathbf{p}_j\}$, $j \in [1, N]$, and a desired mobility type of the mechanism. Our goal is to design a conformable mechanism that conforms to the reference surface S in the stowed state and generates a path/motion that approximates the target 3D path/motion while satisfying the desired mobility type. We formulate designing such a conformable mechanism as an optimization problem, and describe its search space, objective functions, constraints, and our optimization solver.

Search space. The search space of our design problem includes topology and geometry of a conformable mechanism. A conformable mechanism's topology is defined by the number (K) of joints and the type of each joint; see Section 4.1. The parameters that define a conformable mechanism's geometry can be classified into two classes:

1. *Parameters for kinematics.* These parameters consist of the position (u_k, v_k) and orientation Φ_k of each joint J_k in the conforming configuration; see Section 4.2. The mechanism's kinematics is determined by this class of parameters.
2. *Parameters for fabrication.* This class of parameters determines the shape of each joint, each link, and the end-effector. Most of these parameters are predefined such as geometric parameters of each joint type (e.g., radius of the S joint), thickness of the processed reference surface, width of each link and the end-effector. Among all these parameters, we search for the centerline of each link L_i , represented by a spline with control points $\{\mathbf{c}_{i,j}\}$, $j \in [1, m]$, and the centerline of the end-effector, represented by a spline with control points $\{\mathbf{c}_{e,j}\}$, $j \in [1, 5]$; see Section 4.2.

By default, the driving joint J_1 of the mechanism rotates for a whole circle in a motion period. In other words, the joint angle of the driving joint J_1 is within range $[0, 2\pi]$, where 0 means the mechanism is in the conforming configuration. However, in case that the desired mobility type is intramobility or extramobility, the driving joint J_1 cannot rotate for a whole circle. Hence, we need to determine the maximally allowed rotation angle of the driving joint J_1 , denoted by Θ . By this, the range of the driving joint J_1 's joint angle is $[0, \Theta]$.

Objective functions. Among the six requirements in Section 3, the 2nd requirement (i.e., surface conformity) has been satisfied by our geometric modeling of the mechanism in the conforming configuration in Section 4.2, and the 5th requirement (i.e., fabricable mechanism) has been satisfied by our geometry processing of the reference surface in Section 5. We formulate the 3rd (i.e., path/motion generation) and 6th (i.e., mechanism appearance) requirements as two objective functions.

1. *Path/motion approximation function* E_{motion} measures the similarity between the generated path/motion and the target



Fig. 7. (Left) User inputs are a reference surface S , a target 3D path represented by a sequence of 3D points $\{\mathbf{p}_j\}$ (in purple color), and a desired mobility type (transmobility in this example). (Right) Our approach designs a conformable mechanism that conforms to the reference surface S and generates a 3D path (in green color) that approximates the target one.

one. We use dynamic time warping (DTW) algorithm [41] to measure the similarity as:

$$E_{\text{motion}} = DTW[I, J],$$

$$DTW[i, j] = \|\mathbf{M}_e(t_i), \mathbf{p}_j\| + \min \begin{cases} DTW[i-1, j] \\ DTW[i, j-1] \\ DTW[i-1, j-1] \end{cases}, \quad (11)$$

where I is the total number of samples of the motion period T (I is set as 360 in our experiments), $t_i = i \times T/I$ is the sampled time, $\mathbf{M}_e(t_i)$ is the end effector position/pose at time t_i , and \mathbf{p}_j is the j th point/pose in the target path/motion.

2. *Mechanism appearance function* E_{appea} evaluates the mechanism appearance based on E_{gap} in Equation 10. However, purely using E_{gap} tends to minimize gaps in the mechanism at the stowed state via generating mechanisms with long and curly moving links to increase G_M in Equation 10. To avoid this, we penalize moving links that are too long. Thus, the mechanism appearance function is defined as:

$$E_{\text{appea}} = E_{\text{gap}} + \lambda \sum_{i=1}^{K-1} \text{Len}(L_i), \quad (12)$$

where λ is a trade-off factor and is typically set as 0.1 in our experiments.

Constraints. We formulate the 1st requirement (i.e., working mechanism) and the 4th requirement (i.e., mechanism mobility) in Section 3 as constraints.

1. *Linkage kinematics.* According to Equation 4, the mechanism should satisfy the following constraint to ensure validity of its kinematics:

$$\mathbf{C}_L(\mathbf{s}_1, \dots, \mathbf{s}_K) = \mathbf{0}, \quad (13)$$

2. *Singularity-free motion.* To avoid singularities, the minimal distance from the mechanism state $\mathbf{s}(t)$ to the singularities during a whole motion period T should be larger than a threshold μ :

$$\min_{t \in [0, T]} (\sigma_{\min}(\mathbf{B})) \geq \mu, \quad (14)$$

where $\mu = 0.01$ in our experiments.

3. *Collision-free motion.* To avoid collision among moving links, we require a certain distance between centerlines of any pair of moving links:

$$\text{Dist}(\mathbf{x}_i(t), \mathbf{x}_j(t)) > \kappa, \quad 1 \leq i < j \leq K-1, \quad t \in [0, T], \quad (15)$$

where $\mathbf{x}_i(t)$ is the centerline of moving link L_i at time t in the Euclidean space, and κ is the allowed minimal distance between any pair of moving links. In our experiments, κ is set as $2r$, where r is the width of moving links.

4. *Joint angles.* The joint angle of each joint should be within a valid range:

$$\mathbf{s}_k(t) \in \Omega_l, \quad 2 \leq k \leq K, \quad t \in [0, T], \quad (16)$$

where Ω_l is the set of valid joint angle ranges defined for the joint type l in Table 1.

5. *Smooth motion.* We ensure a smooth motion of the mechanism using

$$\max_{t \in [0, T]} \|\mathbf{s}_k(t)'\| < \epsilon, \quad 2 \leq k \leq K, \quad (17)$$

where ϵ is set to 0.03 radian/s in our experiments.

6. *Moving link geometry.* We constrain the centerline of each moving link within the reference surface S at the stowed state (i.e. at time $t = 0$):

$$\tilde{\mathbf{x}}_i(0) \in [0, 1]^2, \quad 1 \leq i \leq K-1, \quad (18)$$

where $\tilde{\mathbf{x}}_i(t)$ is the centerline of moving link L_i in the reference surface S 's UV space.

7. *Mobility type.* The mechanism should satisfy the desired mobility type Mob during the motion period T via

$$O_S(\mathbf{x}_i(t)) = \text{Sign}(\text{Mob}), \quad 1 \leq i \leq K-1, \quad t \in [0, T], \quad (19)$$

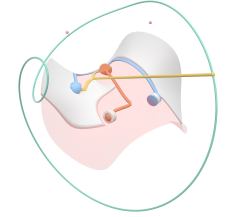
where Mob can be either intramobile or extramobile. According to Equation 7, $\text{Sign}(\text{intramobile}) = -1$ and $\text{Sign}(\text{extramobile}) = 1$.

Optimization solver. The two objectives (Equations 11 and 12) in our optimization problem usually conflict with each other, meaning that approximating the target path/motion better usually degrades the mechanism appearance; see the supplementary material for an example. To solve the multi-objective optimization problem, we employ the widely used Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm to generate a set of Pareto-optimal solutions. Users are allowed to choose a desired result from the set of Pareto-optimal solutions. To explore the search space, we first sample parameters that determine topology and geometry of the mechanism, and then compute the maximally allowed driving joint angle Θ that ensures the desired mobility type in Equation 19. For a conformable mechanism with the sampled topology and geometry and computed Θ , we evaluate the two objective functions (Equations 11 and 12) and verify the above constraints (Equations 13- 18). Please refer to the supplementary material for details of our optimization solver.

Table 2. Statistics of the results shown in this paper. The columns from left to right refers to the reference surface, path or motion generation, number of target points/poses, mechanism topology, values of the two objective functions, and time taken to design each mechanism.

Fig	Surface	Path/ Motion	# points/ poses	Topology	E_{motion}	E_{appea}	Optim. Time (min)
1	Bunny	P	4	RSRU	0.66	0.61	21.4
8	Blob	P	6	RRUS	0.88	0.99	18.4
	Lilium	P	6	RRSU	0.82	0.75	20.1
	Freeholes	P	6	RSRU	0.34	1.04	19.8
	Tetrahedron	P	6	RRSU	1.37	1.9	22.5
10	Flower	P	2	RUSR	0.00009	2.05	13.4
		P	3	RSUR	0.004	0.49	15.7
		P	4	RRUS	0.049	0.72	19.3
		P	5	RSRU	0.077	1.36	20.2
		P	6	RRSU	0.501	1.88	20.9
		P	6	RRSRR	0.143	1.55	25.2
11	Flower	M	2	RSRU	0.0003	0.94	18.7
		M	3	RRURU	0.35	1.73	24.9
12	Igloo	P	6	RRUS	0.79	1.24	19.9
13	Cone	M	2	RRUS	0.02	0.23	18.9
14	Roof	M	2	RSRU	0.12	0.25	24.2
15	Robot Shell	M	2	RRUS	0.08	0.32	17.1

Note that our optimization solver may find some solutions that are not fabricable when the target 3D path has a large spread relative to the reference surface. The inset shows an example, where the fixed joint (i.e., the U joint connecting the orange link) is disconnected from the supporting link, due to the cut of the reference surface by the moving joints/links.



7. Results

We implemented our approach in C++ and libigl [42] on a desktop computer with a 3.0GHz CPU and 32GB memory. We used hpp-fcl [43] for collision detection and CGAL for surface parameterization. Figure 8 shows that our approach is able to design a conformable mechanism that conforms to a reference surface with a variety of shapes, including a surface with three branches, a surface with a cavity in the middle, a surface with three holes, and a closed surface with sharp edges. Figure 9 shows that our approach is able to design a conformable mechanism with a desired mobility type, given the same reference surface. Please refer to the accompanying video for animations of these mechanisms. Table 2 provides statistics of all the results shown in the paper. We observe that both E_{motion} and E_{appea} grows as the number (N) of target points/poses increases.

3D path generation. We conducted an experiment to evaluate the performance of our conformable mechanism for 3D path generation, where we increase the number of 3D points in the target path while fixing the reference surface. In our experiments, we found that our conformable mechanism with four joints is able to generate a 3D path that precisely passes through

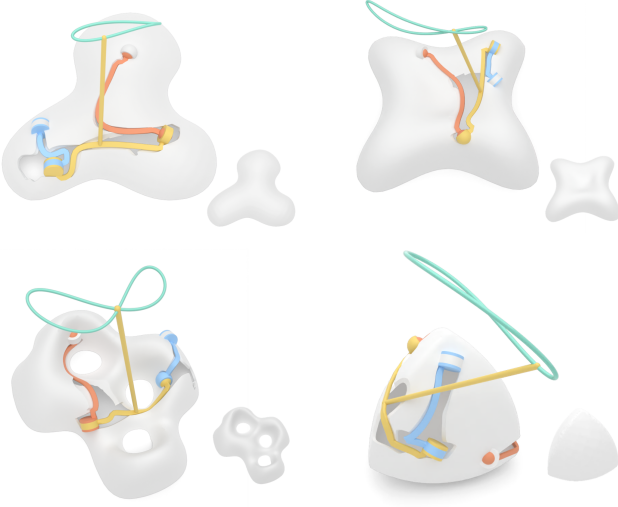


Fig. 8. Our approach allows designing a conformable mechanism that conforms to a reference surface with a variety of shapes.

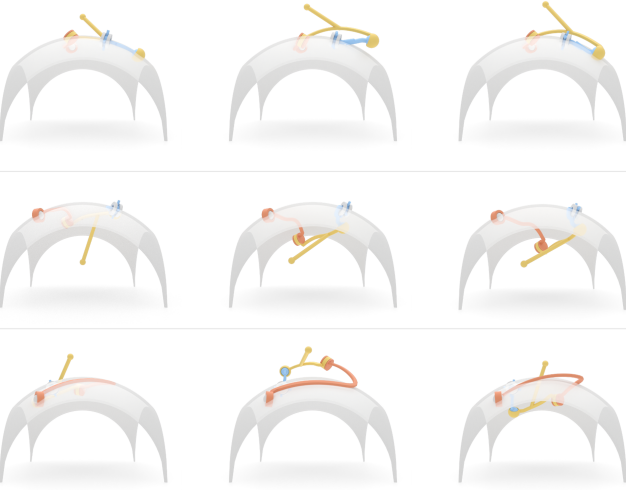


Fig. 9. Our approach allows designing a conformable mechanism with different types of mobility: (top) extramobility, (middle) intramobility, and (bottom) transmobility. We show three configurations for each result, where the first one is the conforming configuration.

2 or 3 target points; see Figure 10(a&b). As the number of target points increases, the 3D path generated by our conformable mechanism with four joints can approximate 4 or 5 target points with small error; see Figure 10(c&d). However, the 3D path generated by our conformable mechanism with four joints cannot closely approximate 6 target points; see Figure 10(e). Introducing one more joint into the conformable mechanism helps to reduce the error for approximating the 6 target points; see Figure 10(f). Please refer to E_{motion} in Table 2 for the path approximation error of each result.

3D motion generation. We conducted an experiment to evaluate the performance of our conformable mechanism for 3D motion generation, where we increased the number of 3D poses in the target motion while fixing the reference surface. In our experiments, we found that our conformable mechanism with four joints is able to generate a 3D motion that precisely follows 2 target poses; see Figure 11(a). As the number of target poses

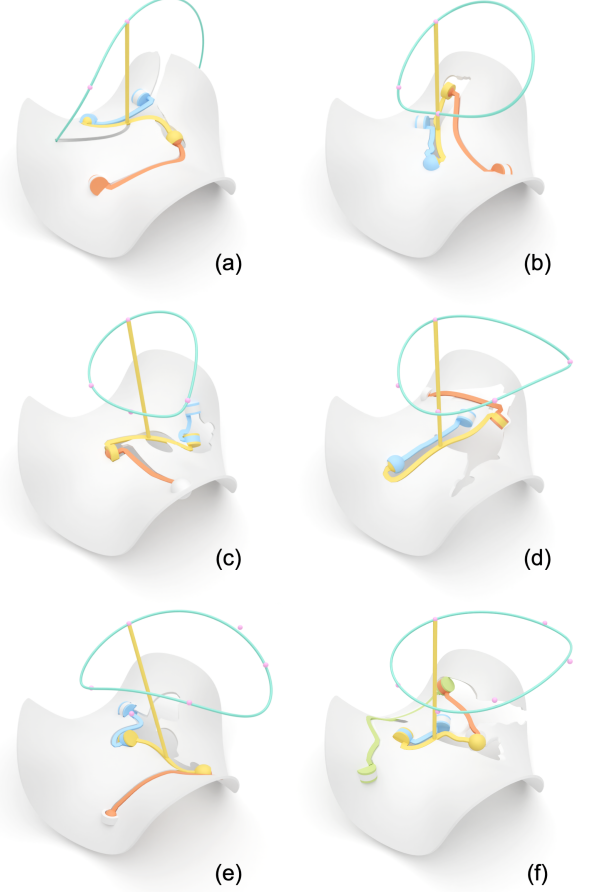


Fig. 10. Given the same reference surface, we design conformable mechanisms with four joints to generate 3D paths represented by (a) 2, (b) 3, (c) 4, (d) 5, and (e) 6 points, respectively. We also design a conformable mechanism with five joints to generate a 3D path represented by (f) 6 points.

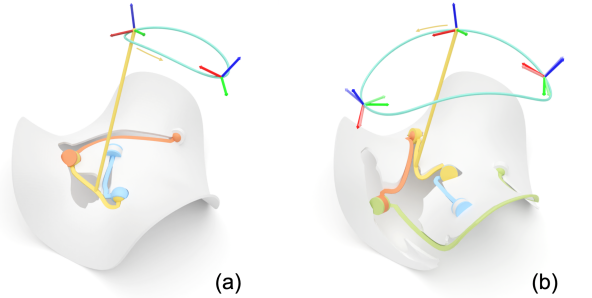


Fig. 11. Given the same reference surface, we design (a) a conformable mechanism with four joints to generate a 3D motion represented by 2 poses and (b) a conformable mechanism with five joints to generate a 3D motion represented by 3 poses. Note that the target and generated poses are rendered in solid and transparent local frames, respectively.

increases, the 3D motion generated by our conformable mechanism with five joints can closely approximate 3 target poses; see Figure 11(b).

Fabrication. To validate the functionality of our conformable mechanism, we used our approach to design a conformable mechanism that conforms to an Igloo surface and generates an eight-like path; see Figure 12(top). To facilitate manual actuation, small handles are added to the driven joint. We fabricated the mechanism as a non-assembly, articulated model (i.e.,

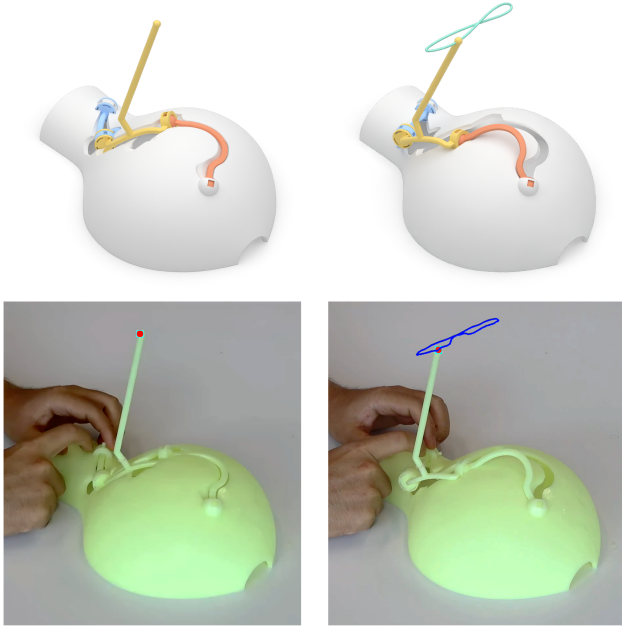


Fig. 12. Validating functionality of our conformable mechanism. (Top) Our designed conformable mechanism in a stowed state and a working state, where the generated path is in green color. (Bottom) The 3D printed mechanism in the corresponding configurations, where the generated path is tracked on video images and visualized in blue.

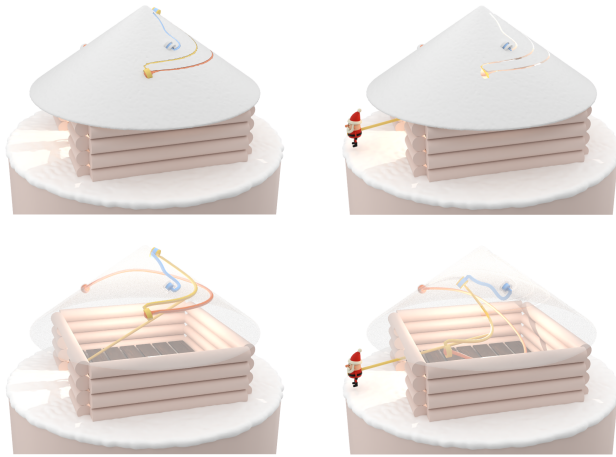


Fig. 13. (Top) Designing a conformable mechanism with intramobility for controlling the motion of a character in a mechanical toy. (Bottom) The toy roof is rendered with transparency to show the mechanism in action.

a single piece) using the Lite SPSS600 printer with UV curable resin material, where the joint tolerance is set as 0.5mm; see Figure 12(bottom). We tracked the path generated by our 3D printed mechanism using an image-based approach. Figure 12 compares our design and the 3D printed prototype in a stowed state and a working state, respectively. We can see that the 3D print matches the virtual design well in the two configurations, and the tracked curve is very close to the generated curve.

Applications. We demonstrate the usefulness of our conformable mechanism with three applications, i.e., mechanical toy, low-cost manipulator, and mobile robot.

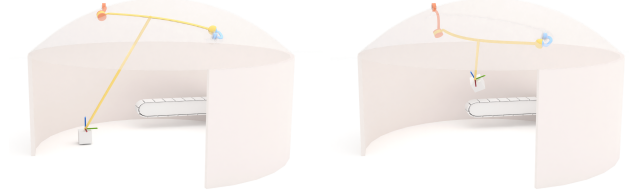


Fig. 14. Designing a conformable mechanism with intramobility as a ceiling-mounted manipulator for performing a pick-and-place task.

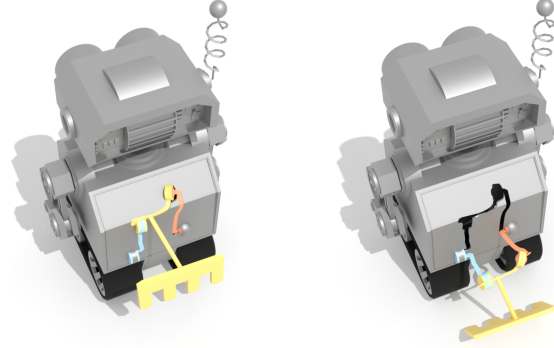


Fig. 15. Designing a conformable mechanism with extramobility for controlling the motion of a tool (in yellow) in a mobile robot for performing agricultural tasks.

- *Mechanical toy.* In this application, we design a conformable mechanism with intramobility for controlling the motion of a character, where the reference surface is the toy shell and the character has two target poses; see Figure 13. Thanks to the surface conformity, our mechanism (in its stowed state) does not occupy any internal space of the toy, which therefore can be used to accommodate the character. After running the toy, our mechanism is able to push and pull the character while being hidden by the toy shell, thanks to intramobility achieved by our approach.
- *Ceiling-mounted manipulator.* In this application, we design a conformable mechanism with intramobility as a ceiling-mounted manipulator for performing a pick-and-place task; see Figure 14. To design the mechanism, the reference surface is the ceiling of a building that the manipulator locates, and the target motion is represented by two poses of the manipulator for picking and placing an item, respectively. The results show that our designed conformable mechanism not only enables the pick-and-place task but also allows the manipulator to be stored in the ceiling when not in use.
- *Mobile robot.* In this application, we design a conformable mechanism with extramobility for controlling the motion of a tool of a mobile robot for performing agricultural tasks; see Figure 15. To design the mechanism, the reference surface is the shell of the robot, and the target motion is represented by two poses: one pose for a stowed tool and one pose for the tool in use. The results show that our designed conformable mechanism is able to control the motion of the tool for performing agricultural tasks and can be stowed in the robot shell when not in use.

8. Conclusion

We have presented conformable mechanisms, a new class of linkage mechanisms whose joints and links conform to a reference surface in one of their configurations. One advantage of conformable mechanisms is that they can be very compact in a stowed state. Our aim is to model and design conformable mechanisms for 3D path and motion generation. To solve this problem, we enumerate topologies of conformable mechanisms, parameterize their geometry, and process the reference surface to ensure fabricability of the mechanism. Taking this as a foundation, we propose an optimization-based approach to design a conformable mechanism for approximating a user-specified 3D path or motion. Our conformable mechanism has been evaluated in many aspects including a variety of freeform surfaces to conform, ability to generate 3D paths and motions, kinematic performance of a 3D printed prototype, and three applications to demonstrate their usefulness.

Limitations and Future work. Our work has several limitations that open up interesting directions for future research. First, modeling the geometry of links beyond their centerlines leads to conformable mechanisms with more general shapes. This generalization could enable designing more visually appealing conformable mechanisms. Second, our current approach only models and designs conformable mechanisms with simple topology, i.e., linkages with a single loop. One future research is to generalize the topology of conformable mechanisms to linkages with multiple loops. Lastly, current conformable mechanisms are designed to approximate a given path/motion but generally cannot exactly reproduce it. This limitation arises from the limited number of design parameters inherent to the mechanism. In this future, we plan to address this limitation by combining our linkage-based conformable mechanisms with cams or non-circular gears [44] for exact 3D path/motion generation.

Acknowledgement

We thank the reviewers for their valuable comments. This work was supported by the MOE AcRF Tier 2 Grant of Singapore (MOE-T2EP20222-0008), the MOE AcRF Tier 1 Grant of Singapore (RG12/22), and the Taith Research Mobility Funding.

References

- [1] McCarthy, JM, Soh, GS. *Geometric Design of Linkages*. Interdisciplinary Applied Mathematics 11, Springer; 2011.
- [2] Nelson, TG, Zimmerman, TK, Magleby, SP, Lang, RJ, Howell, LL. Developable mechanisms on developable surfaces. *Science Robotics* 2019;4(27):eaau5171:1–eaau5171:12.
- [3] Coros, S, Thomaszewski, B, Noris, G, Sueda, S, Forberg, M, Sumner, RW, et al. Computational design of mechanical characters. *ACM Trans on Graph (SIGGRAPH)* 2013;32(4):83:1–83:12.
- [4] Thomaszewski, B, Coros, S, Gauge, D, Megaro, V, Grinspun, E, Gross, M. Computational design of linkage-based characters. *ACM Trans on Graph (SIGGRAPH)* 2014;33(4):64:1–64:9.
- [5] Roussel, R, Cani, MP, Léon, JC, Mitra, NJ. Exploratory design of mechanical devices with motion constraints. *Comp & Graph* 2018;74:244–256.
- [6] Bharaj, G, Coros, S, Thomaszewski, B, Tompkin, J, Bickel, B, Pfister, H. Computational design of walking automata. In: *Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 2015, p. 93–100.
- [7] Megaro, V, Zehnder, J, Bäcker, M, Coros, S, Gross, M, Thomaszewski, B. A computational design tool for compliant mechanisms. *ACM Trans on Graph (SIGGRAPH)* 2017;36(4):82:1–82:12.
- [8] Tang, P, Zehnder, J, Coros, S, Thomaszewski, B. A harmonic balance approach for designing compliant mechanical systems with non-linear periodic motions. *ACM Trans on Graph (SIGGRAPH Asia)* 2020;39(6):191:1–191:14.
- [9] Cheng, Y, Sun, Y, Song, P, Liu, L. Spatial-temporal motion control via composite cam-follower mechanisms. *ACM Trans on Graph (SIGGRAPH Asia)* 2021;40(6):270:1–270:15.
- [10] Cheng, Y, Song, P, Lu, Y, Chew, WJ, Liu, L. Exact 3d path generation via 3d cam-linkage mechanisms. *ACM Trans on Graph (SIGGRAPH Asia)* 2022;41(6):225:1–225:13.
- [11] Zhu, L, Xu, W, Snyder, J, Liu, Y, Wang, G, Guo, B. Motion-guided mechanical toy modeling. *ACM Trans on Graph (SIGGRAPH Asia)* 2012;31(6):127:1–127:10.
- [12] Ceylan, D, Li, W, Mitra, NJ, Agrawala, M, Pauly, M. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans on Graph (SIGGRAPH Asia)* 2013;32(6):186:1–186:11.
- [13] Megaro, V, Thomaszewski, B, Gauge, D, Grinspun, E, Coros, S, Gross, M. ChaCra: An interactive design system for rapid character crafting. In: *Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 2014, p. 123–130.
- [14] Song, P, Wang, X, Tang, X, Fu, CW, Xu, H, Liu, L, et al. Computational design of wind-up toys. *ACM Trans on Graph (SIGGRAPH Asia)* 2017;36(6):238:1–238:13.
- [15] Xu, H, Knoop, E, Coros, S, Bäcker, M. Bend-It: Design and fabrication of kinetic wire characters. *ACM Trans on Graph (SIGGRAPH Asia)* 2018;37(6):239:1–239:15.
- [16] Chen, K, Li, S, Song, P, Zheng, J, Liu, L. mpcMech: Multi-point conjugation mechanisms. *ACM Trans on Graph (SIGGRAPH Asia)* 2024;43(6):211:1–211:14.
- [17] Nishida, G, Bousseau, A, Aliaga, DG. Multi-pose interactive linkage design. *Comp Graph Forum (Eurographics)* 2019;38(2):277–289.
- [18] Bäcker, M, Coros, S, Thomaszewski, B. LinkEdit: Interactive linkage editing using symbolic kinematics. *ACM Trans on Graph (SIGGRAPH)* 2015;34(4):99:1–99:8.
- [19] Pan, Z, Liu, M, Gao, X, Manocha, D. Joint search of optimal topology and trajectory for planar linkages. *The International Journal of Robotics Research* 2023;42(4–5):176–195.
- [20] Nobari, AH, Srivastava, A, Gutfreund, D, Xu, K, Ahmed, F. LInK: Learning joint representations of design and performance spaces through contrastive learning for mechanism synthesis. 2024. ArXiv:2405.20592.
- [21] Sun, J, Liu, W, Chu, J. Synthesis of spherical four-bar linkage for open path generation using wavelet feature parameters. *Mechanism and Machine Theory* 2018;128:33–46.
- [22] Sun, J, Chen, L, Chu, J. Motion generation of spherical four-bar mechanism using harmonic characteristic parameters. *Mechanism and Machine Theory* 2016;95:76–92.
- [23] Chu, J, Sun, J. A new approach to dimension synthesis of spatial four-bar linkage through numerical atlas method. *Journal of Mechanisms and Robotics* 2010;2(4):041004:1–041004:14.
- [24] Liu, W, Sun, J, Chu, J. Synthesis of a spatial rrss mechanism for path generation using the numerical atlas method. *Journal of Mechanical Design* 2020;142(1):012303:1–012303:12.
- [25] Sun, JW, Mu, DQ, Chu, JK. Fourier series method for path generation of rccc mechanism. *Proc of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 2012;226(3):816–827.
- [26] Bai, S, Li, Z, Angeles, J. Exact path synthesis of rccc linkages for a maximum of nine prescribed positions. *Journal of Mechanisms and Robotics* 2022;14(2):021011:1–021011:8.
- [27] Sun, J, Liu, Q, Chu, J. Motion generation of rccc mechanism using numerical atlas. *Mechanics Based Design of Structures and Machines* 2017;45(1):62–75.
- [28] Greenwood, JR, Magleby, SP, Howell, LL. Developable mechanisms on regular cylindrical surfaces. *Mechanism and Machine Theory* 2019;142:103584:1–103584:13.
- [29] Hyatt, LP, Greenwood, JR, Butler, JJ, Magleby, SP, Howell, LL. Methods for mapping mechanisms to developable surfaces. *International*

- Journal of Mechanisms and Robotic Systems 2021;5(1-2):39–60.
- [30] Murray, AP, McCarthy, JM. Computing the developable forms of planar and spherical four-bar linkages. *International Journal of Mechanisms and Robotic Systems* 2021;5(1-2):22–38.
 - [31] Hyatt, LP, Magleby, SP, Howell, LL. Developable mechanisms on right conical surfaces. *Mechanism and Machine Theory* 2020;149:103813:1–103813:13.
 - [32] Butler, J, Greenwood, J, Howell, LL, Magleby, S. Limits of extramobile and intramobile motion of cylindrical developable mechanisms. *Journal of Mechanisms and Robotics* 2021;13(1):011024:1–011024:8.
 - [33] Sheffield, JL, Sargent, B, Howell, LL. Embedded linear-motion developable mechanisms on cylindrical surfaces. *Journal of Mechanisms and Robotics* 2024;16(1):015001:1–015001:11.
 - [34] Vennard, H, Greenwood, J, Butler, J. Design considerations for the dimensional synthesis of cylindrical developable mechanisms. *Journal of Mechanical Design* 2024;146(1):013301:1–013301:11.
 - [35] Uicker, JJ, Pennock, GR, Shigley, JE. *Theory of Machines and Mechanisms* (5th Edition). Oxford University Press; 2016.
 - [36] Eck, M, DeRose, T, Duchamp, T, Hoppe, H, Lounsbery, M, Stuetzle, W. Multiresolution analysis of arbitrary meshes. In: *Proc. of SIGGRAPH*. 1995, p. 173–182.
 - [37] Gosselin, C, Angeles, J. Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation* 1990;6(3):281–290.
 - [38] Zhou, Q, Panetta, J, Zorin, D. Worst-case structural analysis. *ACM Trans on Graph (SIGGRAPH)* 2013;32(4):137:1–137:11.
 - [39] Telea, A, Jalba, A. Voxel-based assessment of printability of 3d shapes. In: *Proc. of Int. Conf. on Mathematical Morphology and Its Applications to Image and Signal Processing*. 2011, p. 393–404.
 - [40] Kimmel, R, Sethian, JA. Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences* 1998;95(15):8431–8435.
 - [41] Müller, M. Dynamic time warping. In: *Information Retrieval for Music and Motion*. Springer Berlin, Heidelberg; 2007, p. 69–84.
 - [42] Jacobson, A, Panozzo, D, et al. libigl: A simple C++ geometry processing library. 2018. <https://libigl.github.io/>.
 - [43] Pan, J, Chitta, S, Manocha, D, Lamiriaux, F, Mirabel, J, Carpentier, J, et al. Hpp-fcl: an extension of the flexible collision library. <https://github.com/humanoid-path-planner/hpp-fcl>; 2015–2023.
 - [44] Xu, H, Fu, T, Song, P, Zhou, M, Fu, CW, Mitra, NJ. Computational design and optimization of non-circular gears. *Comp Graph Forum (Eurographics)* 2020;39(2):399–409.