Kinegami: Algorithmic Design of Compliant Kinematic Chains From Tubular Origami

Wei-Hsi Chen, *Member, IEEE*, Woohyeok Yang, Lucien Peach, Daniel E. Koditschek, *Life Fellow, IEEE*, and Cynthia R. Sung, *Member, IEEE*

***Abstract*—Origami processes can generate both rigid and compliant structures from the same homogeneous sheet material. In this article, we advance the origami robotics literature by showing that it is possible to construct an arbitrary rigid kinematic chain with prescribed joint compliance from a single tubular sheet. Our "Kinegami" algorithm converts a Denavit–Hartenberg specification into a single-sheet crease pattern for an equivalent serial robot mechanism by composing origami modules from a catalogue. The algorithm arises from the key observation that tubular origami linkage design reduces to a Dubins path planning problem. The automatically generated structural connections and movable joints that realize the specified design can also be endowed with independent user-specified compliance. We apply the Kinegami algorithm to a number of common robot mechanisms and hand-fold their algorithmically generated single-sheet crease patterns into functioning kinematic chains. We believe this is the first completely automated end-to-end system for converting an abstract manipulator specification into a physically realizable origami design that requires no additional human input.**

***Index Terms*—Dubins path, kinematic synthesis, origami robot, programmable compliance.**

I. Introduction

ORIGAMI robots are machines whose morphologies and functions are created by folding locally flat sheets [1]. This fabrication and assembly process enables rapid construction of complex 3-D objects and can even incorporate multilayered materials for regional stiffness [2], [3], [4] and circuit integration [5], [6], [7]. Folded modules can be used as joints [8] or near-rigid thin shell structural supports [9]. In this article, we present and analyze an algorithmic pipeline enabling the construction of an entire robot consisting of both rigid links and compliant joints with origami fabrication.

Manuscript received 23 March 2022; revised 13 August 2022; accepted 30 August 2022. Date of publication 10 October 2022; date of current version 5 April 2023. This work was supported in part by the Army Research Office under the SLICE Multidisciplinary University Research Initiatives Program award under Grant W911NF1810327 and in part by the National Science Foundation under Grant 1845339. This article was recommended for publication by Associate Editor C. Rucker and Editor A. Menciassi upon evaluation of the reviewers' comments. *(Corresponding author: Wei-Hsi Chen.)*

The authors are with the General Robotics, Autonomous, Sensing and Perception (GRASP) Lab, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: weicc@seas.upenn.edu; woohyeok@seas.upenn.edu; peach@seas.upenn.edu; kod@seas.upenn.edu; crsung@seas.upenn.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2022.3206711.

Digital Object Identifier 10.1109/TRO.2022.3206711

A. Designing Crease Patterns for Kinematic Mechanisms

The design of origami kinematic structures is more complex than the conventional robot design [10] since it integrates manufacturability with function, requiring 2-D embeddable crease patterns that simultaneously fold into complex spatial shapes [11]. Existing computational methods for automatically generating origami patterns are generally restricted to rigid shapes [12], [13], [14]. In contrast, robot designs require fold patterns that achieve not only the desired geometries but also the desired degrees of freedom. Successful modular approaches [7], [9], [15], [16] to origami robot design involve combining simple patterns for structural links or joints to create more complicated trusses, linkages, and even continuum mechanisms [17]. These approaches leverage libraries of crease patterns generated through in-depth studies of action origami [8], [18], buckling, and bellows patterns [19], [20], [21], and high-degree-of-freedom tessellations of both the origami [22] and kirigami (combining folding and cutting) variety [23], [24]. Partial automation of origami module compositions has been developed for shapes in [25] (joining the crease patterns of the unfolded constituent spatial structures along human specified edges) and to create specific kinematic mechanisms in [16] (allowing their manual joining via discretely constructed hinges or actuators). Other algorithmic work on origami composition [9], [15], including algorithmic resolution of geometric conflicts [7], [16], similarly presumes both prior specification of the modules to be joined and how they should be combined. This article aims to advance the systematic design of origami robots through algorithms that directly translate kinematic specifications into constituent modules and the corresponding crease pattern compositions that fold into the appropriate spatial mechanism.

B. Tunable Compliance in Robots

In addition to kinematic synthesis, a modular design approach provides the opportunity for programmable compliance in the resulting mechanism. There is growing interest in both the analysis [26] and synthesis [21] of origami joints that exploit tunable [27] and parametrizable [28] compliance, as well as the dramatic range of stiffnesses [9] achievable with this hybrid soft-rigid metamaterial. In dynamical tasks, the ability to manage the kinetic and the potential energy of a robot's body and environment allows the robot to manipulate objects using fewer actuated degrees of freedom (DOF) [29] and negotiate otherwise inaccessible environments [30]. Traditionally, dynamical tasks have

required high-power-density actuators [31], [32], [33] where output power is generally proportional to mass [34]. A long-standing tradition of parallel compliance in mechatronics design, when joined to novel, distributable actuation materials [35], can open the door to systematic reduction of actuator mass through the distribution of peak power demand over space and time [36] with consequently increased specific agility [37]. Impressive dynamic behaviors in folded sheet machines have been demonstrated through powerful actuators [38], [39], [40], as well as in transmitting cascaded power into and from compliant folded springs [41] for hopping [42] and leaping [43] behaviors driven by conventional actuators. Again, in all these applications of origami's metamaterial properties (i.e., anisotropically tuned rigid and compliant responses from homogeneous sheets), the desired structural shapes and stiffnesses have emerged from painstakingly planned one-off manually generated designs.

C. Contributions

This article addresses these gaps by providing, to our knowledge, the first completely automated end-to-end system for converting an algebraic mechanism specification into a physically realizable crease pattern that folds to achieve it. Our method follows the rule that no cuts are allowed in the crease pattern so that no additional stress will be introduced around the edge of the cuts. Similar to existing approaches, our system recruits a catalogue of parameterized modules. It provides the additional contribution of automatically choosing the relevant modules and design parameters and composing them into a nonself-intersecting, single-sheet pattern, thus reducing the design problem into one abstract specification. The resulting pipeline does not require additional human input beyond a Denavit–Hartenberg (D-H) specification, though its algorithmic steps are sufficiently transparent to allow the integration of designers' alternative modules or more suitably optimized compositions when desired. Specifically, we present the following.

1) A parameterized catalogue of origami modules for tubular sheets, including rigid connectors and revolute and prismatic joints that exhibit a tunable range of stiffnesses.
2) An algorithm for accessing that catalogue in generating from a D-H specification of a serial robot a kinematically equivalent one-piece origami mechanism.
3) A proof that the algorithm is correct up to the yet unproven claim (Conj. 7) that our spatial link generator (Algorithm 9) will never produce a design that intersects itself.
4) An empirical analysis of how the module parameters determine their corresponding joints' compliance.
5) An empirical demonstration of the efficacy of these algorithmic constructions taking the form of a series of physically realized robot arms.

The rest of this article is organized as fiollows. Section II defines the design problem and proves a comprehensive reader's guide to where each of these contributions is presented. Section III introduces the proposed origami modules that act as building blocks to our algorithm, including their crease patterns and the folded states. Section IV describes how to connect origami modules into origami links. Section V locates origami joints to satisfy a given D-H specification. Section VI synthesizes the previous steps into a complete algorithm that converts an arbitrary D-H specification into a foldable kinematic chain. Section VII presents experimental results, and finally, Section VIII concludes this article.

TABLE I
NOMENCLATURE

Key symbols related to D-H specifications	
$\mathcal{O}_i := \{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i, \boldsymbol{o}_i\}$	the ith joint frame (Sec. II-A1)
$\mathcal{D} := \{a_i, \alpha_i, d_i, \theta_i\}^{N+1}$	D-H specification (Sec. II-A1)
$\boldsymbol{q} \in \mathcal{Q}$	joint states (Sec. II-A1)
$\boldsymbol{T} : \mathbb{R}^3 \to \mathbb{R}^3$	rigid transformation (Sec. II-A1)
$\boldsymbol{R}(\hat{\boldsymbol{e}}, \theta) : \mathbb{R}^3 \to \mathbb{R}^3$	rotational about $\hat{\boldsymbol{e}}$ for θ angle (Sec. III)
Key symbols related to origami modules	
$\mathcal{O} := \{\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{o}\}$	module frames (Sec. III)
$\mathcal{M}, \mathcal{J}, \mathcal{L}, \mathcal{K} := \{\mathcal{V}, \mathcal{E}\}$	module (Sec. II-A2), joint (Sec. III-B), link (Def. 1), and kinematic chain, (Def. 3)
$c(\mathcal{M}), c(\mathcal{L})$	centerline: module (Sec. III), link (Def. 2)
$\boldsymbol{M}, \boldsymbol{L} : \mathbb{R}^3 \to \mathbb{R}^3$	rigid transformation instantiated by a module (Sec. III) and a link (Sec. IV-A)
$\boldsymbol{v}, \boldsymbol{p} \in \mathbb{R}^3$	vertex, reference point
Key symbols related to crease patterns (often "primed" variables)	
$\mathcal{F}, \mathcal{G} := \{\mathcal{V}', \mathcal{E}'\}$	crease pattern of a module (Sec. II-A2), and a link (Sec. IV-A)
$\boldsymbol{F}' : \mathbb{R}^2 \to \mathbb{R}^2$	rigid transformation (Sec. IV-A)
$\boldsymbol{v}', \boldsymbol{p}' \in \mathbb{R}^2$	vertex, reference point
Miscellaneous notation:	
Subscripts: p proximal, d distal (Sec. III), c centroid (Sec. V)	
Barred variables: $\bar{\mathcal{O}}, \bar{\boldsymbol{v}}, \bar{\mathcal{V}}$ homogeneous representation (Sec. III,IV)	

II. Problem Formulation

Given a kinematic description of a serial robot, our goal is to find a crease pattern that folds into a functionally equivalent kinematic chain. The key symbols are summarized in Table I.

A. Definitions

1) Kinematic Chain: A *serial robot*, or a robot with a *kinematic chain* mechanism, is a collection of links and joints where each link is connected by joints to at most two other links. In this work, we follow the convention [45] of using exclusively 1-DOF *revolute joints (R)* and *prismatic joints (P)* and describe the robot using the *D-H convention*. Frames attached to the links track the change in robot posture affected by each joint [46]. We denote a frame $\mathcal{O} = \{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \boldsymbol{o}\}$, where $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ is the orthonormal basis and $\boldsymbol{o}$ is the origin. We use the numbering system in [44] (see Fig. 1) and attach the *joint frame* $\mathcal{O}_i$ for the ith joint to the intersection between the ith joint axis and the common normal of the ith and $(i+1)$th joint axes. Using this convention, the relative position and orientation of the frame $\mathcal{O}_i$ with respect to frame $\mathcal{O}_{i-1}$ can be described with only four parameters: link length a_i, link twist α_i, joint offset d_i, and joint angle θ_i.

The chain's *configuration space* $\mathcal{Q}$ is the set of all possible values of its joint variables and can be written as a Cartesian product of each joint's variable range. Here, we define the joint

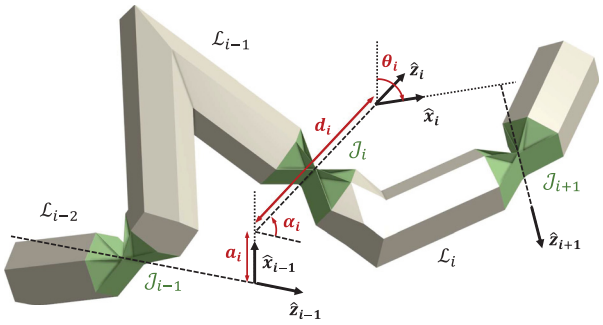


Fig. 1. Schematic of links and joints in the origami kinematic chain and its D-H parameters with the numbering system from [44].

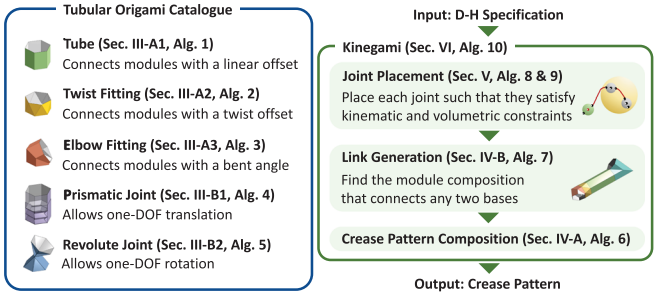


Fig. 2. Tubular origami catalogue and Kinegami algorithm flowchart.

variable as q_i, where the index $i \in \mathbb{N}$ denotes the ith joint. The joint state can be represented as $\boldsymbol{q} = (q_1, q_2, \ldots q_i) \in \mathcal{Q} = \mathbb{R}^i$. We also define the zero configuration $\boldsymbol{q}_0 \in \mathcal{Q}$ of the joint state to be the state where the robot is initialized. For a revolute joint, the joint angle is the joint variable $\theta_i = q_i \in \mathbb{S}^1$. For a prismatic joint, the joint offset is the joint variable $d_i = q_i \in \mathbb{R}$. In practice, these joints could have geometrical limits that only allow them to move on a proper subset of $\mathbb{S}$ or $\mathbb{R}$.

Denote by ${}^{i-1}\boldsymbol{T}_i$ the homogeneous transformation matrix of the $\mathcal{O}_i$ frame relative to the $\mathcal{O}_{i-1}$ frame, as given by the D-H specification $\mathcal{D} := \{a_i, \alpha_i, d_i, \theta_i\}^{N+1}$, where N is the number of joints (and the additional row prescribes the end-effector frame) [47]. Assuming $\mathcal{O}_0$ is the base frame, the forward kinematics for the ith joint is then

$$\mathcal{O}_i = {}^0\boldsymbol{T}_i \mathcal{O}_0, \text{ where } {}^0\boldsymbol{T}_i = {}^0\boldsymbol{T}_1 {}^1\boldsymbol{T}_2 \cdots {}^{i-1}\boldsymbol{T}_i. \quad (1)$$

Note that the D-H specification does not specify the complete morphology of a kinematic chain since the physical location of a joint does not need to coincide with the joint frame. Given $\mathcal{D}$, one can realize more than one kinematic chain that satisfies the specification. We will show in this article one way to construct an origami kinematic chain robot for a given $\mathcal{D}$.

2) Tubular Origami: A *tubular folding sheet* P_T is a sheet that is a spatial embedding of the cylinder, $\mathbb{S}^1 \times [0, 1] \hookrightarrow \mathbb{R}^3$. In other words, P_T is a cylindrical tube in space with "holes" only at either end and no self-intersections. In practice, a tubular sheet can be formed by gluing an opposite pair of a rectangular sheet's edges together. For ease of visualization, this work depicts P_T as a flat rectangle with left and right sides identified. A *crease pattern* is then a partition of the folding sheet P_T into a finite set of open polygons bounded by a set of open line segments bounded by vertices. Each line segment is called a *fold* or a *crease*. Each polygon separated by a cycle of creases and the boundary of the sheet is called a *facet*. We describe a crease pattern as a graph $\mathcal{F} = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \{\boldsymbol{v}'_1, \boldsymbol{v}'_2, \ldots, \boldsymbol{v}'_n\}$, $\boldsymbol{v}'_i = \begin{bmatrix} v'_{i,x} & v'_{i,y} \end{bmatrix}^T$ is the set of vertices and $\mathcal{E}' = \{(i, j) : \boldsymbol{v}'_i, \boldsymbol{v}'_j \in \mathcal{V}'\}$ is the set of edges, or creases. Each crease is associated with a spatial (out of plane) *fold angle* $\mu : \mathcal{E}' \to [-\pi, \pi]$. Creases assigned with negative fold angles are called mountain folds, and creases assigned with positive fold angles are called valley folds. As with the periodic crease pattern tiles in [48], the creases that cross the left and right side borders of P_T must be continuous straight lines with the same fold angle assignments.

Folding is a sequence of rigid transformations consisting of rotating facets by the assigned fold angles about creases. Through folding, each vertex $\boldsymbol{v}'_i$ in the crease pattern is mapped to a location $\boldsymbol{v}_i = \begin{bmatrix} v_{i,x} & v_{i,y} & v_{i,z} \end{bmatrix}^T$ in $\mathbb{R}^3$. The length of the creases and the inner angles of each facet are isometric between the crease pattern and its folded state. Note that this is slightly different from the standard rigid foldability assumptions in that faces may deform during the folding process (and subsequent joint motion), similarly to [48]. We call the final product after folding an *origami module*, denoted as $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ is the set of spatial vertices and $\mathcal{E} = \{(i, j) : \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}\}$ is the set of edges. (Note that we use "primed" variables to differentiate between the $\mathbb{R}^2$ and their corresponding "unprimed" $\mathbb{R}^3$ coordinates.)

3) Joint Compliance: It is sometimes desirable for a joint to exhibit compliance to mediate the dynamical exchange of potential and kinetic energy, such as in a series elastic actuator [49]. This compliance can be intentionally designed into an origami joint by taking advantage of material deformation in the folds and facets [26]. Define the *relaxed configuration* of a spring as the configuration where it stores no energy. Henceforth, this work assumes the zero and relaxed configuration is the same. Then, locally about the relaxed configuration, we can linearize the joints' force-extension relations to get a Hookean stiffness profile $\boldsymbol{k} \in \mathbb{R}^N_{\geq 0}$, or the vector of Hookean stiffnesses of the corresponding N joints.

B. Problem Statement and Structure of Solution

The main contribution of this article is a solution to the following problem.

Problem 1: Given a D-H specification of an N-joint serial robot and its relaxed configuration $\boldsymbol{q}_0$, construct a crease pattern on a tubular sheet P_T that folds into a robot with compliant joints that is kinematically equivalent to the prescribed one.

Our overall strategy to solve this problem is summarized in Fig. 2 and consists of two major parts.

1) Tubular Origami Catalogue: An origami module $\mathcal{M}$ is a 3-D polyhedral shell folded from P_T that instantiates a spatial transformation in $\mathbb{R}^3$ (see Section III). We present a catalogue of modules for translation (see Section III-A1), twisting (see Section III-A2), constant radius turning (see Section III-A3),

1-DOF translation (see Section III-B1), and 1-DOF rotation (see Section III-B2). We find the crease pattern $\mathcal{F}$ of each origami module $\mathcal{M}$ by preserving the isometric constraints between the origami module vertices and the crease pattern vertices.

2) Kinegami Algorithm: A D-H specification $\mathcal{D}$ only describes the frames of the joints, but not their locations. We show how to algorithmically identify the locations of the joints so that they satisfy $\mathcal{D}$ while remaining sufficiently far away from each other (see Section V), such that they can be physically connected using a Dubins-specified link (see Section IV-B)—although the correctness of this particular algorithm remains conjectural (Conj. 7). The crease patterns can be "glued" with no gaps (see Section IV-A) to achieve a single-sheet crease pattern.

Further, we discuss how to specify the stiffnesses of the compliant joints independently:

Problem 2: Given a desired stiffness profile $\boldsymbol{k}$ and an error allowance ϵ, construct a compliant origami robot such that the local stiffness profile around the relaxed configuration $\boldsymbol{q}_0$ lies within $\boldsymbol{k} \pm \epsilon$.

We demonstrate empirically that there are effectively two methods to manipulate the stiffness of an individual joint: increase the effective thickness of the material by using multiple layers (see Section III-B2), or increase facet deformation by manipulating the geometry (see Section III-B1). We defer the formal algorithmic solution to Problem 2 to future work.

III. Origami Modules

We start by constructing the necessary catalogue of origami modules $\mathcal{M}$. To start, P_T is folded into a right prism *origami tube* whose facets are globally flat. The *base*, or the embedded image in $\mathbb{R}^3$ of $\mathbb{S}^1 \times \{0\}$ of the right prism tube, is a regular n_s-sided polygon with a circumradius r. The tube has a total length of h. We define the *proximal* and the *distal* polygon base as the base located at the starting and the ending edge of an origami module, respectively (see Fig. 3). Let the proximal frame $\mathcal{O}_p = \{\hat{\boldsymbol{a}}_p, \hat{\boldsymbol{b}}_p, \hat{\boldsymbol{c}}_p, \boldsymbol{o}_p\}$ and the distal frame $\mathcal{O}_d = \{\hat{\boldsymbol{a}}_d, \hat{\boldsymbol{b}}_d, \hat{\boldsymbol{c}}_d, \boldsymbol{o}_d\}$ lie at the centers of their respective polygonal bases. We define the module's *centerline* as the curve connecting the base centers $\boldsymbol{o}_p$ and $\boldsymbol{o}_d$ along a path that is equidistant to all facets in the module. Then, the "centerline axis" $\hat{\boldsymbol{a}}$ is perpendicular to the base and tangent to the centerline. We choose the axis $\hat{\boldsymbol{b}}$ to lie on the plane of the polygon base and perpendicular to a selected edge. Axis $\hat{\boldsymbol{c}}$ follows the right-hand rule. The frame assignment is not unique since the polygon has rotational symmetry. To simplify the notation of the orientation of the polygon base, we define a reference point $\boldsymbol{p}$ that lies on both the circumference of the polygon and the positive direction on the $\hat{\boldsymbol{b}}$ axis, or

$$\boldsymbol{p} := (r \sin\delta)\hat{\boldsymbol{b}} + \boldsymbol{o} \tag{2}$$

where $\delta = \pi \frac{n_s - 2}{2n_s}$ is half of the interior polygon angle.

Adding or modifying creases to a tubular origami of this form can generate multiple geometries, including *origami fittings* and *origami joints*. An origami fitting is a rigid structure connecting nonintersecting tubes with different base orientations. An origami joint is a mechanism that performs relative motion on the connecting tubes using active folds. In addition, each origami module can be viewed as instantiating a rigid transformation, ${}^p\boldsymbol{M}_d : \mathbb{R}^3 \to \mathbb{R}^3$, that transfers a proximal base to a distal base according to a set of given geometric variables. Fig. 3 shows these origami modules, their spatial transformations, and their crease patterns.

The 3-D shape of the origami module is represented with Cartesian coordinates $\boldsymbol{v}_i$, where $i \in 1, \ldots, n_s$ is the index corresponding to the vertices of the polygon base. Due to the wrap-around effects in indexing, we can define $\boldsymbol{v}_{n_s+1} := \boldsymbol{v}_1$. First, we assign the proximal base vertices $\boldsymbol{v}_{pi}$ recursively through the *polygon base vertex assignments*

$$\begin{aligned} \boldsymbol{v}_{p,1} &:= (r \sin\delta)\hat{\boldsymbol{b}}_p + (r \cos\delta)\hat{\boldsymbol{c}}_p + \boldsymbol{o}_p \\ \boldsymbol{v}_{p,i+1} &:= \boldsymbol{R}(\hat{\boldsymbol{a}}_p, \tfrac{2\pi}{n_s})\,(\boldsymbol{v}_{p,i} - \boldsymbol{o}_p) + \boldsymbol{o}_p \end{aligned} \tag{3}$$

where $\boldsymbol{R}(\hat{\boldsymbol{a}}_p, \frac{2\pi}{n_s})$ is the rotational matrix about the unit vector $\hat{\boldsymbol{a}}_p$ by an angle $2\pi/n_s$. With this assignment, the reference point $\boldsymbol{p}_p$ is on the midpoint of the edge $(\boldsymbol{v}_{p,1}, \boldsymbol{v}_{p,n_s})$. Applying a known rigid transformation ${}^p\boldsymbol{M}_d$ to the proximal base vertices produces the positions of all distal base vertices $\boldsymbol{v}_{d,i}$. For some origami modules, additional vertices are required to capture the full 3-D shape and will be discussed later.

The crease pattern is defined in $\mathbb{R}^2$ where $\boldsymbol{p}'_p := [0 \;\; 0]^T$ is the origin. Intuitively, we can cut the origami module vertically through the reference marker $\boldsymbol{p}_p$ and unwrap the tube into a flat sheet where we align $\boldsymbol{p}_p$ onto $\boldsymbol{p}'_p$. Let $\boldsymbol{v}'_{p,i}$ be the vertices on the crease pattern that correspond to the proximal base vertices $\boldsymbol{v}_{p,i}$ in $\mathbb{R}^3$. Since the vertices $\boldsymbol{v}_{p,i}$ are coplanar, we can assume without loss of generality that the crease pattern vertices $\boldsymbol{v}'_{p,i}$ lie on the x-axis of the crease pattern. And since neighboring vertices of $\boldsymbol{v}_{p,i}$ have the same distance $l_s := 2r \cos\delta$, we can write the *base perimeter vertex assignment* (using modular arithmetic in the subscripts, i, here and henceforth) as

$$\begin{aligned} \boldsymbol{v}'_{p,1} &:= \boldsymbol{p}'_p + \begin{bmatrix} l_s/2 & 0 \end{bmatrix}^T \\ \boldsymbol{v}_{p,i+1} &:= \boldsymbol{v}'_{p,i} + \begin{bmatrix} l_s & 0 \end{bmatrix}^T. \end{aligned} \tag{4}$$

Since $\mathcal{P}_T$ is a developable surface and does not stretch, the corresponding facets between the crease pattern and the origami module are isometric [50]. Thus, the crease pattern vertices, $\boldsymbol{v}'_{d,i}$, that correspond to the distal base vertices, $\boldsymbol{v}_{d,i}$, in the folded state can be found by satisfying isometry constraints. Specifically, the edge between any connected vertices and the angle between any two adjacent edges on a facet must be the same in both the folded state and their crease pattern. All additional crease pattern vertices can be found in this way.

A. Rigid Transformations With Origami Modules

The fundamental building block of the tubular origami modules is the origami tube. We show in this section how to generate both translational and rotational rigid transformations by folding this tube in different ways. Together, the three modules proposed—tube, twist, and elbow fittings—are sufficient to express any rigid 3-D transformation. Section IV-B shows that the problem of designing an origami link can thus be reduced to

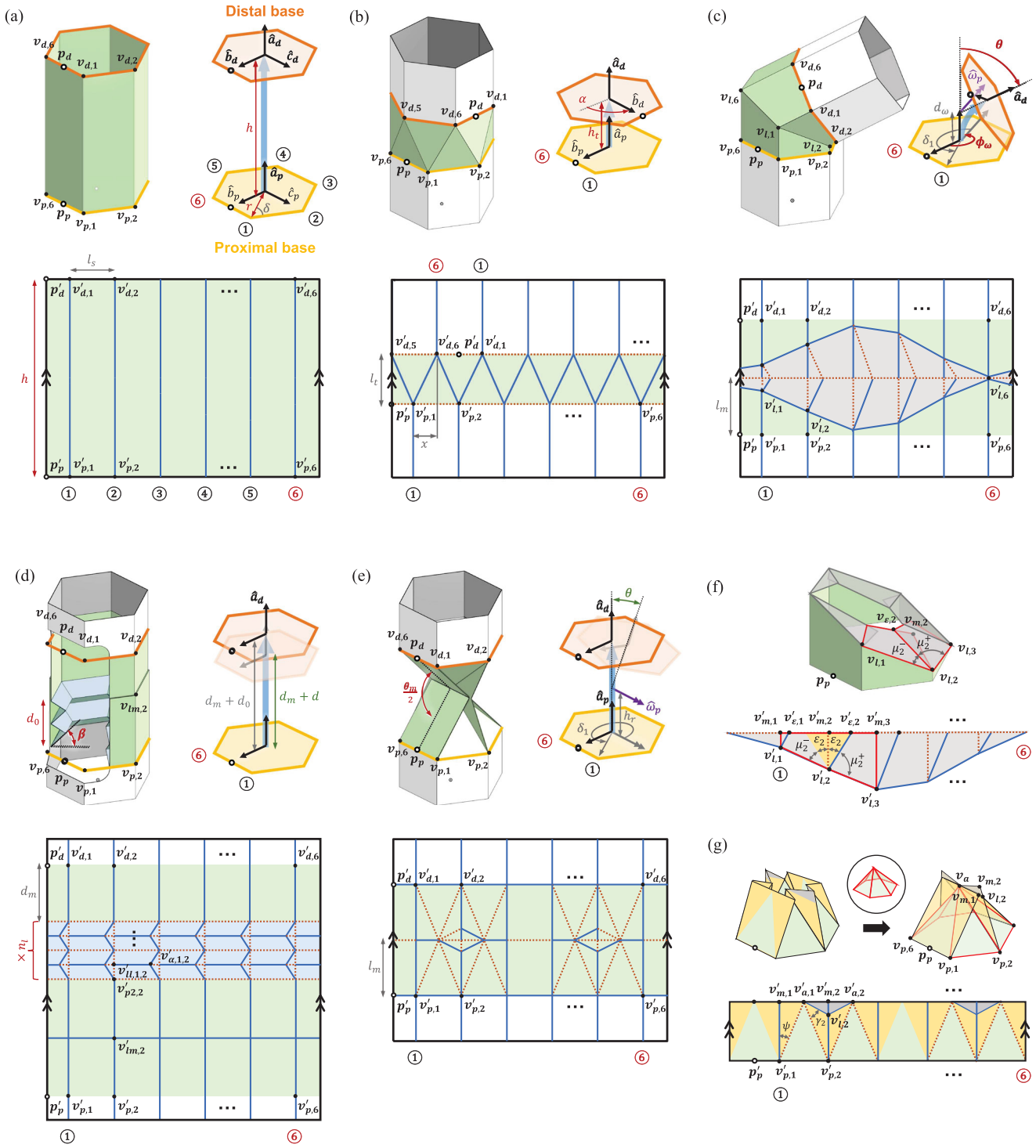


Fig. 3. Folded state of the origami module, its spatial operator representation, and its crease pattern. (a) Origami prism tube in Section III-A1., (b) Twist fitting in Section III-A2. (c) Elbow fitting in Section III-A3. (d) Prismatic joint in Section III-B1. (e) Revolute joint in Section III-B2. In addition, (f) and (g) show the lower half portion of the elbow fitting and the revolute joint, and their closed-up crease patterns, respectively. The actual modules in both the folded states and the crease patterns are colored in green. The crease pattern is drawn on a rectangle with its left and right sides identified, with solid blue lines indicating mountain folds and red dotted lines indicating valley folds. The proximal base of the module is colored yellow and the distal base is orange. The blue arrow that connects the two bases is the centerline of the module. In each case, the input specification (the desired rigid transformation from proximal frame $\mathcal{O}_p$ to distal frame $\mathcal{O}_d$) is represented by red parameters superimposed on the hexagonal base sketch. The polygon circumradius r and the number of sides n_s are inputs. The vertices are marked with numbers in circles and are counted counterclockwise from $\hat{\boldsymbol{b}}$. The origin of the embedded image of $\mathbb{S}^1 \times \{0\}$, or where the axis $\hat{\boldsymbol{b}}$ and the polygon circumference intersect, is marked with p. Notable spatial vertices of the origami modules and their corresponding crease pattern vertices are represented by black parameters superimposed on the figures.

finding a connecting CSC Dubins path [51] and constructing it from these three module types.

1) Translation With the Origami Tube: Translation along the length of the tube is achieved using the basic origami tube, a right regular prism tube of height $h \in \mathbb{R}_{\geq 0}$ with a regular n_s-sided polygon base whose circumradius is r. Fig. 3(a) shows the folded state and crease pattern.

a) Rigid transformation: The origami prism tube instantiates the rigid transformation that translates the proximal base by some distance h along the axis $\hat{\boldsymbol{a}}_p$ to the distal base. The corresponding homogeneous transformation is

$$\bar{\mathcal{O}}_d := {}^p\boldsymbol{M}_d\bar{\mathcal{O}}_p = \begin{bmatrix} \boldsymbol{I} & h\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_p \tag{5}$$

where $\boldsymbol{I}$ is a 3×3 identity matrix, $\boldsymbol{0}$ is a 1×3 array of 0, $\hat{\boldsymbol{e}}_a = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is $\hat{\boldsymbol{a}}_p$ in terms of the body frame $\mathcal{O}_p$, and

$$\bar{\mathcal{O}} = \begin{bmatrix} \hat{\boldsymbol{a}} & \hat{\boldsymbol{b}} & \hat{\boldsymbol{c}} & \boldsymbol{o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

is the 4×4 homogeneous matrix representation of $\mathcal{O}$.

b) Crease pattern: Generating this transformation via tubular origami is straightforward. Intuitively, since the folded state is a right prism, we can cut vertically down one of the edges or facets and unwrap the tube into a flat sheet by laying each rectangular facet next to each other in sequence.

More formally, given the dimensions of the structure—the number of sides n_s, the circumradius r, and the translated distance h—we design the crease pattern shown in Fig. 3(a). The folded state of the origami tube is fully defined by vertices $\mathcal{V} = \{\boldsymbol{v}_{p,i}, \boldsymbol{v}_{d,i}\}$. The vertices of the proximal base $\boldsymbol{v}_{p,i}$ are constrained by the polygon base vertex assignments. The vertices of the distal base $\boldsymbol{v}_{d,i}$ can be found with the homogeneous transformation

$$\bar{\boldsymbol{v}}_{d,i} := \begin{bmatrix} \boldsymbol{I} & h\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \bar{\boldsymbol{v}}_{p,i} \tag{7}$$

where $\bar{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}^T & 1 \end{bmatrix}^T$ is the homogeneous coordinate representation of an affine spatial point.

To map this shape onto a crease pattern, it remains to solve for the vertices $\{\boldsymbol{v}'_{p,i}, \boldsymbol{v}'_{d,i}\}$. The crease pattern variables must satisfy isometry with respect to the folded state variables to constrain their values [48]. The crease pattern vertices that correspond to the proximal base vertices satisfy the base perimeter vertex assignments; thus, $\boldsymbol{v}'_{p,i}$ are fully defined. To find the crease pattern vertices $\boldsymbol{v}'_{d,i}$ that correspond to the distal base, the angles of the facets in the module must be preserved

$$\begin{aligned} &\left(\boldsymbol{v}'_{d,i} - \boldsymbol{v}'_{p,i}\right)^T \left(\boldsymbol{v}'_{p,i+1} - \boldsymbol{v}'_{p,i}\right) \\ &\quad = \begin{bmatrix} v'_{d,i,x} - v'_{p,i,x} & v'_{d,i,y} - v'_{p,i,y} \end{bmatrix} \begin{bmatrix} l_s & 0 \end{bmatrix}^T \\ &\quad = \left(v'_{d,i,x} - v'_{p,i,x}\right) l_s \\ &\quad = \left(\boldsymbol{v}_{d,i} - \boldsymbol{v}_{p,i}\right)^T \left(\boldsymbol{v}_{p,i+1} - \boldsymbol{v}_{p,i}\right) = 0 \end{aligned} \tag{8}$$

Algorithm 1: $Tube(n_s, r, h)$.

Input: polygon shape (n_s, r), and height of the tube h
Output: crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}'_d$

1 $\delta \leftarrow \pi\frac{n_s-2}{2n_s}$; $l_s \leftarrow 2r\cos\delta$;
2 **for** i = 1 **to** n_s **do**
3 $\quad v'_{p,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 0 \end{bmatrix}^T$; $v'_{d,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & h \end{bmatrix}^T$;
4 **end**
5 $\mathcal{F} \leftarrow$ connect vertices with edges shown in Fig. 3 (a);
6 $\boldsymbol{p}'_d \leftarrow \begin{bmatrix} 0 & h \end{bmatrix}^T$;

which simplifies to $v'_{d,i,x} = v'_{p,i,x}$. The edge lengths must match the crease pattern and its folded form, so

$$||\boldsymbol{v}'_{d,i} - \boldsymbol{v}'_{p,i}|| = |v'_{d,i,y} - v'_{p,i,y}| = ||\boldsymbol{v}_{d,i} - \boldsymbol{v}_{p,i}|| = h. \tag{9}$$

Without loss of generality, we always set the distal base vertices to be "above" the proximal base vertices on the crease pattern. We thus find $v'_{d,i,y} - v'_{p,i,y} = v'_{d,i,y} = h$, so $\boldsymbol{v}'_{d,i}$ are also fully defined.

The solution of the crease pattern consists of rectangular panels of height h and width l_s. Algorithm 1 provides the exact vertex locations required to generate the graph. Lines 2–4 locate the crease pattern vertices. In addition, we keep track of the distal marker $\boldsymbol{p}'_p$, the 2-D counterpart to the distal reference point $\boldsymbol{p}_d$ in the folded state (line 6). The x-component of $\boldsymbol{p}'_p$ is the lateral offset from $\boldsymbol{p}'_p$ caused by the twist between the two frames $\mathcal{O}_p$ and $\mathcal{O}_d$ about $\hat{\boldsymbol{a}}_p$, and is 0 for this module. The y-component of $\boldsymbol{p}'_p$ is the height of the crease pattern.

2) Twist With the Twist Fitting: This fitting twists the origami prism tube about $\hat{\boldsymbol{a}}_p$ by an angle $\alpha \in \mathbb{S}^1$, with optional height $h_t \in \mathbb{R}_{\geq 0}$, as shown in Fig. 3(b).

a) Rigid transformation: The twist fitting instantiates the rigid transformation of a screw action on the proximal polygon base with a twist of α and translation h_t along $\hat{\boldsymbol{a}}_p$. The value of h_t can be chosen arbitrarily depending on volume constraints. The corresponding homogeneous transformation is

$$\bar{\mathcal{O}}_d := {}^p\boldsymbol{M}_d\bar{\mathcal{O}}_p = \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_a, \alpha) & h_t\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_p. \tag{10}$$

Due to the rotational symmetry of a regular polygon, the distal polygon transformed after a twist angle of α is the same as the one transformed with a twist angle of $\bar{\alpha} = \alpha \bmod (2\pi/n_s)$. To ensure the origami module is a convex antiprism, we take advantage of the rotational symmetry and rotate the distal *base* by only $\bar{\alpha}$, but rotate the distal *frame* by α.

b) Crease pattern: Inspired by the triangulated cylinder geometry [20], this crease pattern folds into a convex antiprism for structural stability. Given the polygon base parameters n_s and r, the twist angle α with respect to the axis $\hat{\boldsymbol{a}}$, and the height h_t, we design the crease pattern shown in Fig. 3(b).

The folded state of the twist fitting consists of proximal base vertices $\boldsymbol{v}_p$ that satisfy the polygon base vertex assignments, and

Algorithm 2: $TwistFitting(n_s, r, \alpha, h_t)$.

Input: polygon shape (n_s, r), the twist angle α and height of the twist fitting h_t

Output: crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}_d'$

1 $\delta \leftarrow \pi\frac{n_s-2}{2n_s}$; $l_s \leftarrow 2r\cos\delta$;

2 $\overline{\alpha} \leftarrow \alpha \bmod \left(\frac{2\pi}{n_s}\right)$;

3 $x \leftarrow \left(1-\cos\overline{\alpha}+\cot\frac{\pi}{n_s}\sin\overline{\alpha}\right)\frac{l_s}{2}$;

4 $l_t \leftarrow \sqrt{h_t^2+\left(l_s\csc\frac{\pi}{n_s}\sin\left(\frac{\pi}{n_s}-\frac{\overline{\alpha}}{2}\right)\sin\frac{\overline{\alpha}}{2}\right)^2}$;

5 **for** $i = 1$ **to** n_s **do**

6 $\quad v_{p,i}' \leftarrow \begin{bmatrix}(i-1/2)l_s & 0\end{bmatrix}^T$;

7 $\quad v_{d,i}' \leftarrow \begin{bmatrix}((i-1/2)l_s+x) \bmod n_s l_s & l_t\end{bmatrix}^T$;

8 **end**

9 $\mathcal{F} \leftarrow$ connect vertices with edges shown in Fig. 3 (b);

10 $\boldsymbol{p}_d' \leftarrow \begin{bmatrix}\lfloor\frac{n_s\alpha}{2\pi}\rfloor l_s+x & l_t\end{bmatrix}^T$;

the distal base vertices $\boldsymbol{v}_d$ that meet the transformation

$$\bar{\boldsymbol{v}}_d := \begin{bmatrix}\boldsymbol{R}(\hat{\boldsymbol{e}}_a, \bar{\alpha}) & h_t\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1\end{bmatrix}\bar{\boldsymbol{v}}_p. \tag{11}$$

The crease pattern can be found by solving the constraints, where $\boldsymbol{v}_{p,i}'$ satisfy the base perimeter vertex assignments, and $\boldsymbol{v}_{d,i}'$ satisfy the isometry constraints, as shown in Fig. 3(c)

$$\begin{aligned}||\boldsymbol{v}_{d,j}' - \boldsymbol{v}_{p,i}'|| &= ||\boldsymbol{v}_{d,j} - \boldsymbol{v}_{p,i}|| \\ ||\boldsymbol{v}_{d,j}' - \boldsymbol{v}_{p,i+1}'|| &= ||\boldsymbol{v}_{d,j} - \boldsymbol{v}_{p,i+1}||\end{aligned} \tag{12}$$

where $j = (i - \lfloor\frac{n_s\alpha}{2\pi}\rfloor) \bmod n_s$ is the index of the closest distal polygon vertex to the ith proximal vertex "counting about $\hat{\boldsymbol{a}}_p$," and $\lfloor\cdot\rfloor$ is the floor operator. The two sets of equations, along with the polygon side length $||\boldsymbol{v}_{p,i+1}' - \boldsymbol{v}_{p,i}'|| = l_s$, define the vertices of a triangle given three known side lengths. As the distal vertices must be above the proximal vertices $v_{d,j,y}' > v_{p,i,y}'$, it now follows that $\boldsymbol{v}_{d,i}'$ are uniquely defined.

The steps to generate the graph are shown in Algorithm 2. The pattern essentially consists of folds connecting the vertices of the offset proximal and distal prisms into a series of adjacent triangular facets. The offset x in terms of pattern distance is a function of the twist angle α as calculated in line 3. Line 4 calculates the height l_t of the crease pattern from the height of the module h. The resulting pattern has a size of $n_s l_s \times l_t$. The twist angle sets a perimeter offset between the $\boldsymbol{p}_p'$ and $\boldsymbol{p}_d'$, and the x-component of the $\boldsymbol{p}_p'$ can be computed as

$$p_{d,x}' = \lfloor\tfrac{n_s\alpha}{2\pi}\rfloor l_s + x. \tag{13}$$

3) Rotation With the Elbow Fitting: The elbow fitting bends an origami prism tube by an angle $\theta \in (-\pi, \pi)$ along an arbitrary axis $\hat{\boldsymbol{w}}_p$ (described in the proximal frame $\mathcal{O}_p$, where the angle between $\hat{\boldsymbol{w}}_p$ and $\hat{\boldsymbol{b}}_p$ is ϕ_ω) that is perpendicular to the centerline $\hat{\boldsymbol{a}}_p$, as shown in Fig. 3(c). Elbow fittings with larger θ can be achieved by composing n smaller elbow fittings with θ/n together (see Section IV for composition).

a) Rigid transformation: The 3-D shape corresponding to the fitting is essentially two truncated prism tubes joined at an angle of θ to each other. This rotation requires a minimum distance $d_w = r\tan\frac{\theta}{2}$, which depends on the rotational axis $\hat{\boldsymbol{w}}_p$, the bending angle θ, and the circumradius r. An elbow fitting can thus be represented as imposing a rigid transformation that translates the proximal base along $\hat{\boldsymbol{a}}_p$ for a distance d_w, rotates it along the axis $\hat{\boldsymbol{w}}$ by an angle θ, then translates along $\hat{\boldsymbol{a}}_d$ for another distance d_w. The corresponding homogeneous transformation is

$$\bar{\mathcal{O}}_d := {}^p\boldsymbol{M}_d\bar{\mathcal{O}}_p = \begin{bmatrix}\boldsymbol{R}(\hat{\boldsymbol{w}}_p, \theta) & d_w\left(\boldsymbol{R}(\hat{\boldsymbol{w}}_p, \theta) + \boldsymbol{I}\right)\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1\end{bmatrix}\bar{\mathcal{O}}_p. \tag{14}$$

The centerline of the elbow fitting was initially defined as the two line segments intersecting at the rotational axis. Observe that (14) gives the same rigid transformation of a *constant radius turn* with radius r for an angle θ about $\hat{\boldsymbol{w}}_p$ located at the instant center of rotation, $\boldsymbol{o}_{icr} := \mathrm{sgn}(\theta)r\boldsymbol{R}(\hat{\boldsymbol{a}}_p, \frac{\pi}{2})\hat{\boldsymbol{w}}_p + \boldsymbol{o}_p$. Henceforth, we redefine the *centerline of an elbow fitting* to be a circular arc centered at $\boldsymbol{o}_{icr}$ with radius r and angle θ.

b) Crease pattern: The pattern is inspired by Gieseking's Crimp-bent tubes [48], [52]. Given the polygon base parameters n_s and r, the rotational axis angle ϕ_w of $\hat{\boldsymbol{w}}_p$ with respect to $\hat{\boldsymbol{b}}_p$, and the bending angle θ, we construct the crease pattern shown in Fig. 3(c). Only half of the module is described, since it is symmetric about a center plane. The vertices and edges of the other half can then be generated through mirroring.

As shown in Fig. 3(c) and (f), the pattern essentially consists of two parts: the green facets that are exposed on the outside and the gray facets that are hidden away on the interior of the tube. The exterior faces form a truncated prism with vertices $\{\boldsymbol{v}_{p,i}, \boldsymbol{v}_{l,i}\}$. The vertices of the proximal polygon base $\boldsymbol{v}_p$ follow the polygon base vertex assignments. The top face, depicted in Fig. 3(f), is a distorted polygon formed by the intersection of the tube and a plane oriented at an angle $\theta/2$ relative to the proximal base. The distorted polygon vertices $\boldsymbol{v}_{l,i}$ are located at the intersection of the vertical line from the corresponding proximal base vertices and the angled plane

$$\begin{aligned}v_{l,i,x} &:= v_{p,i,x} \ \text{ and } \ v_{l,i,y} := v_{p,i,y} \\ \left(\boldsymbol{R}\left(\hat{\boldsymbol{w}}_p, \tfrac{\theta}{2}\right)\hat{\boldsymbol{a}}_p\right)^T\left(\boldsymbol{v}_{l,i} - (d_w\hat{\boldsymbol{a}}_p + \boldsymbol{o}_p)\right) &= 0.\end{aligned} \tag{15}$$

Since $\theta \in (-\pi, \pi)$, $d_w = r\tan\frac{\theta}{2} \in \mathbb{R}$, and (15) always has a unique solution.

Now we find the crease pattern vertices $\boldsymbol{v}_{p,i}'$ and $\boldsymbol{v}_{l,i}'$. The proximal vertices $\boldsymbol{v}_{p,i}'$ are defined with the base perimeter vertex assignments (4). To preserve the isometry of the edge and the angle of the truncated prism, we have

$$\begin{aligned}\left(\boldsymbol{v}_{l,i}' - \boldsymbol{v}_{p,i}'\right)^T\left(\boldsymbol{v}_{p,i+1}' - \boldsymbol{v}_{p,i}'\right) &= \left(\boldsymbol{v}_{l,i} - \boldsymbol{v}_{p,i}\right)^T\left(\boldsymbol{v}_{p,i+1} - \boldsymbol{v}_{p,i}\right) \\ &= 0\end{aligned} \tag{16}$$

$$||\boldsymbol{v}_{l,i}' - \boldsymbol{v}_{p,i}'|| = ||\boldsymbol{v}_{l,i} - \boldsymbol{v}_{p,i}||. \tag{17}$$

Identical arguments to those for (8) and (9) (i.e., the vertices $\boldsymbol{v}_{l,i}'$ are directly above $\boldsymbol{v}_{p,i}'$ on the crease pattern) now yield a complete specification of $\boldsymbol{v}_{l,i}'$.

Since the top boundary of exposed faces forms an irregular shape, the remaining material in the tube [gray shaded area

in Fig. 3(f)] must be *tucked* away to the interior of the tube. At a high level, the tucking operation consists of adding extra creases to the unwanted additional sheet material and folding it to the inside of the polyhedral surface [53]. The tucks in this pattern are triangular, as shown in Fig. 3(f). The purpose of these tucks is to reduce the angle between the crease pattern edges $\angle \boldsymbol{v}'_{l,i-1}\boldsymbol{v}'_{l,i}\boldsymbol{v}'_{l,i+1}$ such that the sum of the remaining angles μ_i^- and μ_i^+ equals the interior angle of the corresponding vertices $\angle \boldsymbol{v}_{l,i-1}\boldsymbol{v}_{l,i}\boldsymbol{v}_{l,i+1}$ of the distorted polygon in the folded state. In other words, the tucks will allow us to fold the (gray) extra sheet flat on the center plane that splits the elbow joint. We thus fold the extra material away by introducing two folds: one at the edge of the triangular material being tucked and one at the bisector of the triangle. These two folds bring the two yellow triangles onto each other in the folded state, resulting in the edges $(\boldsymbol{v}'_{l,i-1}, \boldsymbol{v}'_{l,i})$ and $(\boldsymbol{v}'_{l,i}, \boldsymbol{v}'_{l,i+1})$ lying at an angle of $\mu_i^- + \mu_i^+$ apart. The triangular tuck can be placed at any orientation as long as the sum of $\mu_i^- + \mu_i^+$ remains the same, so we choose to place the bisector vertically in the crease pattern. Thus, the vertex $\boldsymbol{v}'_{m,i}$ is at

$$\boldsymbol{v}'_{m,i} := \begin{bmatrix} v'_{l,i,x} & \max_j v'_{l,j,y} \end{bmatrix}^T \tag{18}$$

and an additional crease $(\boldsymbol{v}'_{p,i}, \boldsymbol{v}'_{\varepsilon,i})$ is placed at an angle from the crease $(\boldsymbol{v}'_{p,i}, \boldsymbol{v}'_{m,i})$, where its vertex $\boldsymbol{v}'_{m,i}$ is located directly on the right-hand side of $\boldsymbol{v}'_{m,i}$, or

$$\begin{aligned} \varepsilon_i &:= \left(\angle \boldsymbol{v}'_{l,i-1}\boldsymbol{v}'_{l,i}\boldsymbol{v}'_{l,i+1} - \angle \boldsymbol{v}_{l,i-1}\boldsymbol{v}_{l,i}\boldsymbol{v}_{l,i+1}\right)/2 \\ \boldsymbol{v}'_{\varepsilon,i} &:= \begin{bmatrix} v'_{l,i,x} + \left(v'_{m,i,y} - v'_{l,i,y}\right)\tan\varepsilon_i & v'_{m,i,y} \end{bmatrix}^T. \end{aligned} \tag{19}$$

Algorithm 3 contains the precise steps used to generate the resulting graph. Specifically, lines 2–10 calculate the position of the vertices for the crease pattern. The resulting pattern has a size of $n_s l_s \times 2l_m$. Since $l_m \propto \tan\frac{\theta}{2}$, the larger the bending angle for an elbow fitting is, the longer the tubular sheet is needed for the construction. Now, observe the following.

Remark One can split a desired bending angle θ into n portions by constructing n identical elbow fittings with rotational angle $\theta/n \leq \pi/2$ and composing the n elbow fittings using Algorithm 6. In doing so, the total length of the tubular sheet is reduced. In fact, it is possible to achieve larger bending angles $2\pi \geq |\theta| \geq \pi$ by splitting the bending angle.

B. Parameterized Rigid Transformation With Origami Joints

Origami joints allow the two rigid bodies they connect to move relative to each other. We denote the origami joint as $\mathcal{J}$, a particular instance of $\mathcal{M}$. We propose two 1-DOF lower pair joints folded out of tubular sheets: the prismatic and revolute joints. These two joints are sufficient to compose any desired kinematic motion [45]. Each of the proposed origami joints has an initial state where all the facets are flat. When the joint activates, the folding angles in the origami module change, and sometimes its facets deform. Creases that change angle during the joint motion are called *active folds*. By manipulating the active folds, we can change the stiffness profiles of the joints.

Algorithm 3: $ElbowFitting(n_s, r, \theta, \phi_w)$.

Input: Polygon shape (n_s, r), bending angle θ, and rotational axis angle ϕ_w
Output: Crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}'_d$

1 $\delta \leftarrow \pi\frac{n_s-2}{2n_s}$; $l_s \leftarrow 2r\cos\delta$;
2 $d_w \leftarrow r\tan\frac{\theta}{2}$;
3 **for** $i = 1$ **to** n_s **do**
4 $\quad \delta_i \leftarrow \pi\frac{2i-1}{n_s} - \phi_w$; $l_i \leftarrow d_w(\sin\delta_i + 1)$;
5 **end**
6 $l_m \leftarrow \max l_i$;
7 **for** $i = 1$ **to** n_s **do**
8 $\quad \boldsymbol{v}'_{p,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 0 \end{bmatrix}^T$; $\boldsymbol{v}'_{d,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 2l_m \end{bmatrix}^T$;
9 $\quad \boldsymbol{v}'_{l,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & l_i \end{bmatrix}^T$; $\boldsymbol{v}'_{m,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & l_m \end{bmatrix}^T$;
10 **end**
11 $\mathcal{F} \leftarrow$ Connect vertices with edges shown in Fig. 3 (c);
12 Find the crease pattern to tuck away the hidden section [gray shaded area in Fig. 3 (c) and (f)];
13 $\boldsymbol{p}'_d \leftarrow \begin{bmatrix} 0 & 2l_m \end{bmatrix}^T$.

1) 1-DOF Translation With the Prismatic Joint: The prismatic joint is based on a REBO spring [21], [42], an origami pattern that can store potential energy into both its facets and folds when compressed. By changing the cone angle β, the layer height h_0, and the number of layers n_l, a REBO will exhibit tunably varied Hookean stiffness (i.e., an approximately linear selectable force-extension relation) around its zero configuration $d_0 \in \mathbb{R}_{\geq 0}$. Given the desired relaxed configuration length d_0, the single-layer height of REBO can be found as $h_0 = d_0/n_l$. The maximum range of motion of the prismatic joint is $d_m = n_l l_l$, where $l_l = \frac{h_0}{2}\csc\beta$.

To constrain the REBO spring from bending radially, we construct a *wall* surrounding it, limiting its motion to axial translation. The wall is a double-layered prism tube constructed with a crimp fold [54] arranged horizontally to the crease pattern of a tube, as shown in the lower green area of Fig. 3(d). It has a length of d_m, so the entire range of motion is constrained. The joint state $d + d_m$ is the distance between the proximal and distal bases of the prismatic joint, where $d \in [0, d_m]$. An example of the origami prismatic joint, rigid transformation, and its crease pattern is shown in Fig. 3(d).

a) Rigid transformation: The prismatic joint is a function of the joint variable d and translates the proximal base for a distance of $d + d_m$. The homogeneous transformation is

$$\bar{\mathcal{O}}_d := {}^p\boldsymbol{M}_d(d)\bar{\mathcal{O}}_p = \begin{bmatrix} \boldsymbol{I} & (d_m + d)\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix}\bar{\mathcal{O}}_p. \tag{20}$$

b) Crease pattern: Given the polygon base parameters n_s and r, relaxed configuration length d_0, number of layers n_l, and the cone angle β, we construct the crease pattern shown in Fig. 3(d) with three segments: a REBO spring, a wall, and an additional inner tube. We start with the REBO structure colored in blue in Fig. 3(d). Its crease pattern is a function of the polygon shape parameters n_s and r, the cone angle β of the origami bellows, and the layers n_l, and is fully defined in [41]. In addition, we build a wall that is d_m long to constrain the REBO

Algorithm 4: $PrismaticJoint(n_s, r, d_0, n_l, \beta)$.

Input: Polygon shape (n_s, r), zero configuration length d_0, number of layers n_l, cone angle β

Output: Crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}'_d$

1 $\delta \leftarrow \pi\frac{n_s-2}{2n_s}$; $l_s \leftarrow 2r\cos\delta$;
2 $h_0 \leftarrow \frac{d_0}{n_l}$; $l_l \leftarrow \frac{h_0}{2}\csc\beta$; $d_m \leftarrow 2n_l l_l$;
3 $\varphi \leftarrow \frac{\pi}{2}\left(1-2\cos\frac{\beta}{n_s}\right)$;
4 **for** $i = 1$ **to** n_s **do**
5 $\quad \boldsymbol{v}'_{p,i} \leftarrow \begin{bmatrix}(i-1/2)l_s & 0\end{bmatrix}^T$; $\boldsymbol{v}'_{d,i} \leftarrow \begin{bmatrix}(i-1/2)l_s & 4d_m\end{bmatrix}^T$;
6 $\quad \boldsymbol{v}'_{lm,i} \leftarrow \begin{bmatrix}(i-1/2)l_s & d_m\end{bmatrix}^T$; $\boldsymbol{v}'_{p2,i} \leftarrow \begin{bmatrix}(i-1/2)l_s & 2d_m\end{bmatrix}^T$;
7 $\quad$ **for** $j = 1$ **to** n_l **do**
8 $\quad\quad \boldsymbol{v}'_{ll,i,j} \leftarrow \boldsymbol{v}'_{p2,i} + \begin{bmatrix}0 & (2j-1)l_l\end{bmatrix}^T$;
9 $\quad\quad \boldsymbol{v}'_{\alpha,i,j} \leftarrow \boldsymbol{v}'_{p2,i} + \begin{bmatrix}-l_l\cot\varphi & (2j-1)l_l\end{bmatrix}^T$;
10 $\quad\quad \boldsymbol{v}'_{p3,i,j} \leftarrow \boldsymbol{v}'_{p2,i} + \begin{bmatrix}0 & 2jl_l)\end{bmatrix}^T$;
11 $\quad$ **end**
12 **end**
13 $\mathcal{F} \leftarrow$ Connect vertices with edges shown in Fig. 3 (d);
14 $\boldsymbol{p}'_d \leftarrow \begin{bmatrix}0 & 4d_m\end{bmatrix}^T$.

structure. Its crease pattern is shown in the lower green area of Fig. 3(d). We also need an additional inner slider, which is just an origami prism tube that is d_m long. Its crease pattern is shown in the upper green area of Fig. 3(d). The entire crease pattern is a sequence of crease patterns of tubes and the REBO structure, as shown in Fig. 3(d).

Algorithm 4 contains the steps to generate the crease pattern of the prismatic joint. The main parameters are the height h_0 of each layer from the zero configuration, the maximal distance of the motion d_m, and the angle of the diagonal fold ψ (lines 2 and 3). The resulting pattern has a size of $n_s l_s \times 4d_m$.

c) Stiffness: The translational movement of the origami prismatic joint is due to the deformation of the facets. We experimentally validate that the stiffness of the REBO spring is a function of the cone angle β in [21]. Since the origami prismatic joint contains the same REBO spring, its stiffness can also be programmed with the cone angle.

2) 1-DOF Rotation With the Revolute Joint: The origami revolute joint is similar to the *hinge joint* in [24] and allows the tubes to rotate freely by an angle $\theta \in [-\frac{\theta_m}{2}, \frac{\theta_m}{2}]$ about the axis $\hat{\boldsymbol{w}}_p = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ (described from $\mathcal{O}_p$) that is parallel to $\hat{\boldsymbol{c}}_p$ and perpendicular to $\hat{\boldsymbol{a}}_p$. The angle limit $\theta_m \in [0, 2\pi)$ is a design variable that determines the total range of motion of this joint. An example is shown in Fig. 3(d). In addition, we can tune the stiffness of the revolute joint by adding additional crease patterns, as shown in Fig. 4(a).

a) Rigid transformation: Here, the revolute joint is created after pinching the walls of the tube to form active folds at the axis of rotation $\hat{\boldsymbol{w}}_p$. When the joint moves, the angle between $\hat{\boldsymbol{a}}_p$ and $\hat{\boldsymbol{a}}_d$ changes, and the angle is defined as the joint variable $\theta \in [-\frac{\theta_m}{2}, \frac{\theta_m}{2}]$. The limit of the achievable rotational motion occurs when the two neighboring rectangle facets perpendicular to $\hat{\boldsymbol{w}}_p$ touch each other. Thus, the rigid transformation is a function of the joint state θ and can be written as the homogeneous

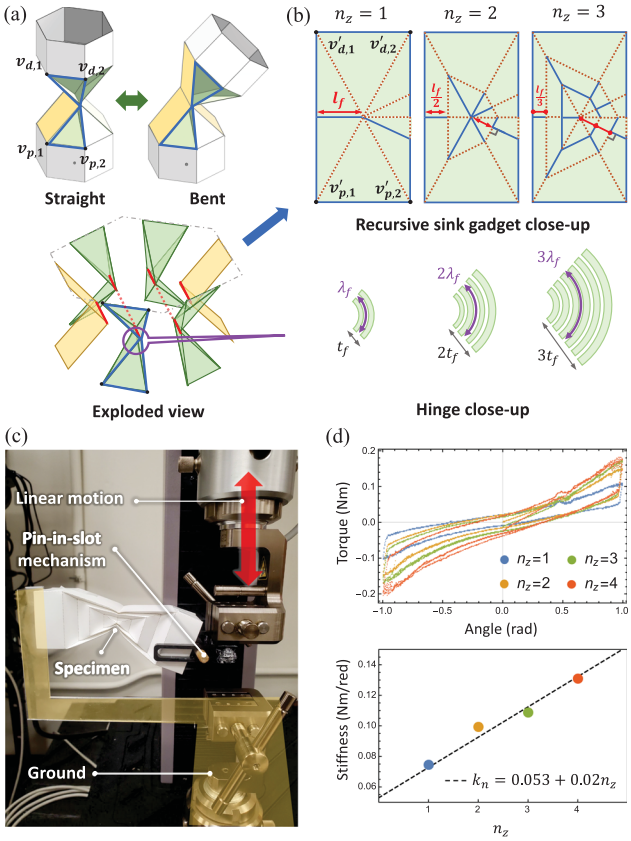


Fig. 4. Stiffness of the revolute joint. (a) Revolute joint in the straight and bent configuration on top, and the exploded view of the facets of the revolute joint on the bottom, where the red creases indicate the active folds. (b) Recursive sink gadget of $n_z = 1, 2, 3$. The close-up of the interested crease pattern area $(\boldsymbol{v}'_{p,1}, \boldsymbol{v}'_{p,2}, \boldsymbol{v}'_{d,2}, \boldsymbol{v}'_{d,1})$ is shown on top, and the hinge around the rotational axis of the revolute joint is shown at the bottom. The effective fold length l_f is marked on the partial crease pattern. In the hinge close-up, the green solid arc represents the cross-section of the layered sheet; t_f is the effective thickness of the stacked layers, and λ_f is the length of the small-length flexural pivot. (c) Experimental setup to measure the stiffness of the revolute joints. (d) The top chart shows the experimental mechanical test for an origami revolute joint design with $n_s = 4$, $r = 2\sqrt{2}$, $\theta_m = \frac{4\pi}{3}$. Different numbers of recursive sink gadget layers are applied $n_z = 1, 2, 3, 4$. The specimen is compressed and pulled between ± 1 rad for three cycles. The bottom chart shows the linear-fitted stiffness from the previous plot concerning n_z with a high correlation of $R^2 = 0.99$.

transformation

$$\bar{\mathcal{O}}_d := {}^p\boldsymbol{M}_d(\theta)\bar{\mathcal{O}}_p = \begin{bmatrix}\boldsymbol{R}(\hat{\boldsymbol{w}}_p, \theta) & h_r\left(\boldsymbol{R}(\hat{\boldsymbol{w}}_p, \theta) + \boldsymbol{I}\right)\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1\end{bmatrix}\bar{\mathcal{O}}_p \tag{21}$$

where $h_r = r\sin\delta\tan\frac{\theta_m}{4} \in \mathbb{R}_{>0}$ is the distance from the proximal (or distal) frame to the rotational axis and is a function of the desired range of motion θ_m.

b) Crease pattern: This pattern is inspired by the commercially familiar milk carton design, but its base is generalized to any even-sided ($n_s \in 2\mathbb{N}$) regular polygon. Given the even-sided polygon base parameters n_s and r and the maximum range of rotation θ_m, we construct the crease pattern shown in Fig. 3(d). In addition, the stiffness of the revolute joint can be programmed by adding the *recursive sink gadget*, shown in Fig. 4(a) and (b) with the boundary edges of $(\boldsymbol{v}'_{p,i}, \boldsymbol{v}'_{p,i+1}, \boldsymbol{v}'_{d,i+1}, \boldsymbol{v}'_{d,i})$. A *gadget*

is a local graph that replaces an existing patch with the same boundary edges to add functionality or modify the pattern [55].

The origami revolute joint can be thought of as two n_s-gon pyramids whose apex vertices are connected, with article flaps to constrain its relative motion. Since the module is symmetric about the center plane, we again describe only half of the pattern here. Fig. 3(g) shows an instance of pinching the walls of the tube to form creases at the axis of rotation in the center and shows that the origami structure fully covers a n_s-gon pyramid. The pyramid formed by the green facets, as shown in the circled subfigure in Fig. 3(g), is fully defined with the vertices of the proximal polygon base $\boldsymbol{v}_p$ (which follow the polygon base vertex assignments), and the apex vertex $\boldsymbol{v}_a$ as

$$\boldsymbol{v}_a := h_r \hat{\boldsymbol{a}}_p + \boldsymbol{o}_p. \tag{22}$$

The remaining material (highlighted in yellow) joins together at the axis $\hat{\boldsymbol{w}}_p$ to constrain the rotation of the pyramids, or

$$\boldsymbol{v}_{m,j}, \boldsymbol{v}_{l,k} \in \{\boldsymbol{v} | \boldsymbol{v} = t\hat{\boldsymbol{w}}_p + \boldsymbol{v}_a, t \in \mathbb{R}\} \tag{23}$$

where $j = 1, \frac{n_s}{2}, \frac{n_s}{2}+1, n_s$ and $k \neq j$. Additionally, the vertices $\boldsymbol{v}_{m,i}$, and $\boldsymbol{v}_{l,j}$ also need to satisfy the angle constraints

$$\angle \boldsymbol{v}_a \boldsymbol{v}_{m,i} \boldsymbol{v}_{p,i} = \tfrac{\pi}{2} \;\; \forall i \tag{24}$$

$$\angle \boldsymbol{v}_a \boldsymbol{v}_{p,j} \boldsymbol{v}_{l,j} = \psi \;\; \forall k \neq j \tag{25}$$

where $\psi = \operatorname{atan2}(l_s, 2l_m)$ is the bisector angle of the yellow triangular flap shown in Fig. 3(g). Since this active fold is a line, it has 1 DOF that allows revolution of the structure. Thus, all vertices for the flaps are defined.

The crease pattern of this half module consists of vertices $\boldsymbol{v}'_p$, $\boldsymbol{v}'_m$, $\boldsymbol{v}'_a$, and $\boldsymbol{v}'_l$. The crease pattern's proximal vertices $\boldsymbol{v}'_p$ are defined with the base perimeter vertex assignments. The height of the half crease pattern is the height of the triangle of the n_s-gon pyramid, thus

$$v'_{m,i,y} = v'_{a,i,y} = l_m = r \sin\delta \sec\tfrac{\theta_m}{4}. \tag{26}$$

In addition, the x-coordinate of $\boldsymbol{v}'_m$ and $\boldsymbol{v}'_a$ (apex of the isosceles triangle) can be easily found to be

$$v'_{m,i,x} := v'_{p,i,x} \tag{27}$$

$$v'_{a,i,x} := (v'_{p,i,x} + v'_{p,i,x})/2. \tag{28}$$

From (23) to (25), we can see that $||\boldsymbol{v}_{l,i} - \boldsymbol{v}_{p,i}|| \leq ||\boldsymbol{v}_{m,i} - \boldsymbol{v}_{p,i}||$, showing that triangle $\triangle \boldsymbol{v}_a \boldsymbol{v}_{p,i} \boldsymbol{v}_{l,i}$ is always covered by $\triangle \boldsymbol{v}_a \boldsymbol{v}_{p,i} \boldsymbol{v}_{m,i}$, and thus, there is extra sheet from the yellow flaps that extend out to the other side of the center plane, as shown in the gray area in Fig. 3(g). Since the half module and its mirrored module have the same amount of extra material, we can tuck the extra material (see *crimp fold* form [54]). To construct this crease, we can find the vertex $\boldsymbol{v}'_{l,i}$ once we find the angle $\gamma_i := \angle \boldsymbol{v}_{p,i} \boldsymbol{v}_a \boldsymbol{v}_{l,i}$.

Algorithm 5 contains the steps used to generate the crease pattern. The main parameters of interest are half the height of the origami joint h_r, half the height of the crease pattern l_m, and angle ψ (lines 2 and 3). The crease pattern vertices are calculated in lines 4–15. In particular, line 14 constructs the recursive sink gadget pattern shown in Fig. 4(b), which essentially consists of concentric polygons centered around the joint axis. The resulting pattern has a size of $n_s l_s \times 2l_m$.

Algorithm 5: $RevoluteJoint(n_s, r, \theta_m, n_z)$.

Input: polygon shape (n_s, r), total bending angle θ, number of recursive sink gadget layers n_z
Output: crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}'_d$

1 $\delta \leftarrow \pi\frac{n_s-2}{2n_s}$; $l_s \leftarrow 2r\cos\delta$;
2 $h_r \leftarrow r\sin\delta\tan\frac{\theta_m}{4}$; $l_m \leftarrow r\sin\delta\sec\frac{\theta_m}{4}$;
3 $\psi \leftarrow \operatorname{atan2}(l_s, 2l_m)$;
4 **for** i = 1 **to** n_s **do**
5 $\quad v'_{p,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 0 \end{bmatrix}^T$; $v'_{d,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 2l_m \end{bmatrix}^T$;
6 $\quad v'_{m,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & l_m \end{bmatrix}^T$; $v'_{a,i} \leftarrow \begin{bmatrix} il_s & l_m \end{bmatrix}^T$;
7 **end**
8 **for** $i \neq 1, \frac{n_s}{2}, \frac{n_s}{2}+1, n_s$ **do**
9 $\quad \delta_i \leftarrow \pi\left(\frac{n_s-4i+2}{2n_s}\right)$;
10 $\quad \gamma_i \leftarrow \operatorname{atan2}\left(\sqrt{h_r^2 + (r\sin\delta_i)^2}, r\cos\delta_i\right)$;
11 $\quad l_i \leftarrow \sqrt{h_r^2 + (r\sin\delta_i)^2}\sec\left(\frac{\pi}{2} - \gamma_i - \psi\right)$;
12 $\quad v'_{l1,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & l_i \end{bmatrix}^T$;
13 $\quad v'_{l2,i} \leftarrow \begin{bmatrix} (i-1/2)l_s & 2l_m - l_i \end{bmatrix}^T$;
14 $\quad$ generate vertices and connect edges for the recursive sink gadget shown in Fig. 4 (b);
15 **end**
16 $\mathcal{F} \leftarrow$ connect vertices with edges shown in Fig. 3 (e);
17 $\boldsymbol{p}'_d \leftarrow \begin{bmatrix} 0 & 2l_m \end{bmatrix}^T$.

c) Stiffness: The revolute joint is a hinge made with stacks of sheet materials. Fig. 4(a) shows the exploded view of a revolute joint with $n_s = 6$, consisting of two sets of yellow front and back hinge units and $(n_s - 2)$ sets of green side hinge units. The active folds that contribute to the revolute motion are colored in red. Since all the active folds experience the same deformation during a bending motion, we can treat them as a system of torsional springs in parallel. The total stiffness is the sum of the individual ones. Assume the stiffness of the front or back hinge (yellow) unit is k_0 and the stiffness of the side hinge (green) unit is $k_{g,1}$. The stiffness of the side hinge unit can be modeled as a small-length flexural pivot [26] as

$$k_{g,1} = K_\Theta E I_f {\lambda_f}^{-1} \tag{29}$$

where K_Θ is the nondimensionalized stiffness, E is Young's modulus, and λ_f is the length of the small-length flexural pivot. The area moment of inertia is $I_f = l_f t_f^3/12$, where l_f is the effective length and t_f is the effective thickness of the active fold. Fig. 4(b) shows an example of the geometric parameters. The overall torsional stiffness of a basic revolute joint K_1 is then

$$K_1 = (n_s - 2)k_{g,1} + 2k_0 = K_{g,1} + K_0 \tag{30}$$

where $K_{g,1} = (n_s - 2)k_{g,1}$ and $K_0 = 2k_0$ for simplicity.

The stiffness of the revolute joint can be programmed by adding n_z sets of recursive fold patterns, or the recursive sink gadget, to the joints, as shown in Fig. 4(b). The recursive sink gadget is inspired by the origami combination fold known as the *closed sink*. A closed sink is a simple inversion of a coned vertex formed from a region in the interior of a sheet, turning a coned mountain peak into a valley and vice versa about a plane. The

crease of the sink lies on the described plane and runs around the point being sunk like a road girdling a mountain peak (or valley). All the fold angles of the creases enclosed by the sink line are converted to the opposite sign [54]. We can increase the sets n_z of recursive sink creases surrounding the same internal vertex and sequentially fold them to create multiple sinks.

When $n_z = 1$, the recursive sink gadget is the same as the side hinge unit described earlier, and we have the original revolute joint. When the recursive sink gadget with $n_z \geq 2$ is introduced, the effective thickness of the active fold increases discretely as $n_z t_f$ and the effective active fold length decreases as l_f/n_z. Observe that as n_z increases, the total length of the neutral line of the active fold increases to $n_z \lambda_f$, as illustrated in Fig. 4(b). Thus, the stiffness of each recursive sink gadget with n_z layers can be written as

$$k_{g,n} = K_\Theta E \left(\tfrac{1}{12} \left(n_z^{-1} l_f \right) \left(n_z t_f \right)^3 \right) \left(n_z \lambda_f \right)^{-1} = n_z k_{g,1} \tag{31}$$

and the total stiffness of the joint is

$$\begin{aligned} K_n &= (n_s - 2) k_{g,n} + 2k_0 = (n_s - 2) n_z k_{g,1} + 2k_0 \\ &= n_z K_{g,1} + K_0. \end{aligned} \tag{32}$$

To show that the stiffness of the revolute joint is programmable, we performed a bending test on the mechanical testing station (MTS Criterion C41 with 1-kN load cell), as shown in Fig. 4(c). The origami revolute joints use parameters $n_s = 4$, $r = 2\sqrt{2}$, $\theta_m = \frac{4\pi}{3}$ and are folded out of perforated 8 mil thick Durilla synthetics paper with polyester finish (CTI Paper, USA). We tested the specimen with different numbers of recursive sink gadget layers $n_z = 1, 2, 3, 4$. To fully constrain all the active folds to align in a straight line, we tied a fishing line around the hinge so that all the active folds are approximately coaligned. The specimen was compressed and pulled between ± 1 rad three times.

The results for the four specimens are shown in the upper subplot of Fig. 4(d). At the beginning of the test, each specimen exhibited a Hookean torsion spring behavior. As the cycle progresses, we start to see hysteresis in every specimen, similar to that observed in [24]. Although there is hysteresis, each specimen has reliable behavior, and the torque–angle curve aligns well for the three cycles. We fit a linear regression to the entire cycle of each curve to find its average slope (torsional stiffness). The results are summarized in the lower subplot of Fig. 4(d). A linear fit indicates that the torsional stiffness K_n increases as an affine function $K_n = 0.053 + 0.02 n_z$ with $R^2 = 0.99$, and thus, verifies our prediction that the stiffness is a linear function of n_z with $K_0 = 0.053$.

C. Expanding the Tubular Origami Catalogue

Although we propose only five modules in our tubular origami catalogue, additional tubular modules can easily be included in our design approach, provided their corresponding rigid transformations are well characterized. For instance, the Kresling pattern studied in [17] is a 3-DOF origami joint that enables sideways bending perpendicular to and a screw motion along the centerline, and it can be added to the catalogue as a higher order joint pair.

IV. Composing Origami Modules Into Links

The origami modules in Section III can be composed to form *links* that instantiate the more general class of rigid transformations between a desired proximal and distal base. This section shows that such a relationship can be realized by following a spatial Dubins path, assuming a sufficient separation between their origins. We develop a module composition scheme and show how to factor any rigid transformation that satisfies this sufficient condition into a product of rigid transformations that such compositions can algorithmically instantiate.

A. Sequential Composition of Origami Modules

More formally, consider joining two tubular origami modules $\mathcal{M}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{M}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ to generate more complex structures. Identifying the homogeneous matrix $\boldsymbol{M}$ with a rigid transformation $\mathbb{R}^3 \to \mathbb{R}^3$, we write that a module $\mathcal{M}_i$ has been transformed by $\boldsymbol{M}$ with abuse of notation as

$$\boldsymbol{M}\mathcal{M}_i := (\boldsymbol{M}\mathcal{V}_i, \mathcal{E}_i) \tag{33}$$

that is, every vertex in $\mathcal{M}_i$ is transformed by the homogeneous transformation matrix $\boldsymbol{M}$ (and the edges follow). We write that two modules have been merged as a union of the graphs

$$\mathcal{M}_1 \cup \mathcal{M}_2 := (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2) \tag{34}$$

where two vertices are considered the same if they occupy the same coordinates in $\mathbb{R}^3$. With these two operations, we define the *module composition* of $\mathcal{M}_2$ onto $\mathcal{M}_1$ as

$$\mathcal{M}_1 \uplus \mathcal{M}_2 := \mathcal{M}_1 \cup \left({}^{1p}\boldsymbol{M}_{1d} \mathcal{M}_2 \right) \tag{35}$$

where ${}^{1p}\boldsymbol{M}_{1d}$ is the rigid transformation instantiated by the module $\mathcal{M}_1$ that translates its proximal base to its distal base. That is to say, $\mathcal{M}_2$ composed onto $\mathcal{M}_1$ is equivalent to $\mathcal{M}_1$ merged with a transformed $\mathcal{M}_2$ such that the distal end of $\mathcal{M}_1$ is coincident with the proximal end of $\mathcal{M}_2$. Note that this operation is not commutative.

Using the module composition, we can define an origami link, or link $\mathcal{L}$ for short, as follows.

Definition 1 (Origami link): A link is the composition of one or more tubes, twist fittings, or elbow fittings of the same base.

The superscript k denotes the number of modules of a composed link. In the base case, a link $\mathcal{L}^1$ is equivalent to a single origami module $\mathcal{M}_1$ that is one of a tube, a twist fitting, or an elbow fitting. If $\mathcal{L}^{k-1}$ and $\mathcal{M}_k$ have the same polygon base shape, we form the link $\mathcal{L}^k$ by composing the two said links in order. In other words

$$\mathcal{L}^1 := \mathcal{M}_1 \text{ and } \mathcal{L}^k := \mathcal{L}^{k-1} \uplus \mathcal{M}_k. \tag{36}$$

The proximal and distal frames of a link $\mathcal{L}^k$ are denoted $\mathcal{O}_p$ and $\mathcal{O}_d$, respectively, such that its proximal frame is $\mathcal{M}_1$'s proximal frame $\mathcal{O}_p = \mathcal{O}_{1p}$ and its distal frame is $\mathcal{M}_k$'s distal frame $\mathcal{O}_d = \mathcal{O}_{kd}$. Observe that the rigid transformation instantiated by a link is the product of those instantiated by its constituent modules, as we now state formally:

Lemma 1 (Rigid transformation of a link): Given a link $\mathcal{L}^k$, the rigid transformation relating the distal to proximal base frames, denoted as $\boldsymbol{L}^k$, is given by

$$\boldsymbol{L}^k = \prod_{j=1}^{k} {}^{jp}\boldsymbol{M}_{jd}. \tag{37}$$

Proof: We prove this by induction. First, consider the base case where $k = 1$. $\mathcal{L}^1$ is simply a module $\mathcal{M}_1$, and the transformation matrix corresponding to the linkage $\mathcal{L}^1$ is

$$\boldsymbol{L}^1 = {}^{1p}\boldsymbol{M}_{1d}. \tag{38}$$

For the inductive step, we assume that (37) is true for link $\mathcal{L}^k$. Now consider the link $\mathcal{L}^{k+1} = \mathcal{L}^k \uplus \mathcal{M}_{k+1}$, which is the composition of a module $\mathcal{M}_{k+1}$ onto an existing link $\mathcal{L}^k$. Based on (35), the module $\mathcal{M}_{k+1}$ is transformed by $\boldsymbol{L}^k$, which corresponds precisely to the placement of the link's distal base frame. By placing $\mathcal{M}_{k+1}$ so that its proximal frame coincides with the distal frame of $\mathcal{L}^k$, we conclude that

$$\boldsymbol{L}^{k+1} = \left(\boldsymbol{L}^k\right)\left({}^{(k+1)p}\boldsymbol{M}_{(k+1)d}\right) = \prod_{j=1}^{k+1} {}^{jp}\boldsymbol{M}_{jd}. \tag{39}$$

■

Assume $\mathcal{F}_i$ is the crease pattern that folds into $\mathcal{M}_i$. Corresponding to (35), we define the *crease pattern composition* with abuse of notation as

$$\mathcal{F}_1 \uplus \mathcal{F}_2 := \mathcal{F}_1 \cup \left({}^{1p}\boldsymbol{F'}_{1d}\mathcal{F}_2\right) \tag{40}$$

where ${}^{1p}\boldsymbol{F'}_{1d}$ is the rigid transformation $\mathbb{R}^2 \to \mathbb{R}^2$ that shifts the crease pattern such that the proximal origin $\boldsymbol{p}'_{2p}$ of $\mathcal{F}_2$ is coincident to the distal marker $\boldsymbol{p}'_{1d}$ of $\mathcal{F}_1$, or

$${}^{1p}\boldsymbol{F'}_{1d} := \begin{bmatrix} \mathbf{I} & \boldsymbol{p}'_{1d} - \boldsymbol{p}'_{1p} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{41}$$

The graph union is computed with any vertices occupying the exact $\mathbb{R}^2$ coordinates being considered identical. With crease pattern composition, we define a crease pattern $\mathcal{G}^k$ recursively by composing the individual crease pattern $\mathcal{F}_j$ of the module $\mathcal{M}_j$ in the order of the module composition of $\mathcal{L}^k$ as

$$\mathcal{G}^1 := \mathcal{F}_1 \text{ and } \mathcal{G}^k := \mathcal{G}^{k-1} \uplus \mathcal{F}_k. \tag{42}$$

Now we show that the link $\mathcal{L}^k$ composed with Def. 1 has no hole or gap, and that $\mathcal{G}^k$ folds into this link. We will find it convenient to denote by $\mathcal{V}_j^p \subset \mathcal{V}_j$ (respectively, $\mathcal{V}_j^d \subset \mathcal{V}_j$) the spatial vertices of the proximal (respectively, distal) base polygons of $\mathcal{M}_j$.

Lemma 2 (Sequential composition of tubular origami): The crease pattern of link $\mathcal{L}^k$ is $\mathcal{G}^k$.

Proof: Denote by ${}^j\bar{\mathcal{V}}_j := \begin{bmatrix} {}^j\bar{\mathcal{V}}_j^p & {}^j\bar{\mathcal{V}}_j^d \end{bmatrix}$ the array of homogeneous vector representations of the (proximal and distal) base vertices of the module $\mathcal{M}_j$ with respect to its proximal frame. Now, from the constructions of Section III, observe that

$${}^j\bar{\mathcal{V}}_j^p = \bar{\mathcal{V}}_0 \text{ and } {}^j\bar{\mathcal{V}}_j^d = {}^{jp}\boldsymbol{M}_{jd}{}^j\bar{\mathcal{V}}_j^p = {}^{jp}\boldsymbol{M}_{jd}\bar{\mathcal{V}}_0 \tag{43}$$

Algorithm 6: $ComposeCP((\mathcal{F}_1, \boldsymbol{p}'_{1d}), (\mathcal{F}_2, \boldsymbol{p}'_{2d}))$.

Input: two crease patterns in an ordered list with their distal marker $((\mathcal{F}_1, \boldsymbol{p}'_{1d}), (\mathcal{F}_2, \boldsymbol{p}'_{2d}))$
Output: crease pattern $\mathcal{F}$, distal marker $\boldsymbol{p}'_d$
1: $\mathcal{F} \leftarrow \mathcal{F}_1 \uplus \mathcal{F}_2$;
2: $\boldsymbol{p}'_d \leftarrow (\boldsymbol{p}'_{1d} + \boldsymbol{p}'_{2d}) \bmod n_s l_s$;

where $\bar{\mathcal{V}}_0$ is the homogeneous matrix representation of the base polygon vertices with respect to the base frame given in (3). For the link $\mathcal{L}^1 = \mathcal{M}_1$, we can write

$${}^1\bar{\mathcal{V}}_1 = \begin{bmatrix} \bar{\mathcal{V}}_0 & {}^{1p}\boldsymbol{M}_{1d}\bar{\mathcal{V}}_0 \end{bmatrix}. \tag{44}$$

Similarly, for the link $\mathcal{L}^k$, denote by ${}^1\bar{\mathcal{V}}_k := \begin{bmatrix} {}^1\bar{\mathcal{V}}_k^p & {}^1\bar{\mathcal{V}}_k^d \end{bmatrix}$ the array of the homogeneous vector representations of the (proximal and distal) base vertices of the included module $\mathcal{M}_k$ with respect to the proximal frame of $\mathcal{L}^k$. It follows that

$$\begin{aligned} {}^1\bar{\mathcal{V}}_k &= \boldsymbol{L}^{k-1}\,{}^k\bar{\mathcal{V}}_k = \boldsymbol{L}^{k-1}\begin{bmatrix} \bar{\mathcal{V}}_0 & {}^{kp}\boldsymbol{M}_{kd}\bar{\mathcal{V}}_0 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{L}^{k-1}\,\bar{\mathcal{V}}_0 & \boldsymbol{L}^k\,\bar{\mathcal{V}}_0 \end{bmatrix} \end{aligned} \tag{45}$$

exhibiting that the proximal base vertices of $\mathcal{M}_k$ coincide exactly with the distal base vertices of $\mathcal{M}_{k-1}$, or $\mathcal{V}_{k-1}^d = \mathcal{V}_k^p$. Thus, the two polygon bases are identified and there are no seams or gaps between the two modules.

We now use this result to prove the lemma using induction. For the base step, it is trivial that $\mathcal{G}^1 = \mathcal{F}_1$ folds into $\mathcal{L}_1 = \mathcal{M}_1$. For the induction step, assume $\mathcal{G}^j$ is the crease pattern that folds into $\mathcal{L}^j$, and $\mathcal{F}_{j+1}$ folds into $\mathcal{M}_{j+1}$. Now consider the crease pattern $\mathcal{G}^{j+1} = \mathcal{G}^j \uplus \mathcal{F}_{j+1}$, which is the composition of a crease pattern $\mathcal{F}_{j+1}$ onto the existing crease pattern $\mathcal{G}^j$ such that the proximal crease pattern vertices $\mathcal{V}'^p_{j+1}$ of $\mathcal{F}_{j+1}$ are identified with the distal crease pattern vertices $\mathcal{V}'^d_j$ of $\mathcal{G}^j$. Since the crease pattern base vertices must correspond to the same spatial vertices, it now follows that $\mathcal{V}_{k-1}^d = \mathcal{V}_k^p$. Thus, $\mathcal{G}^{j+1}$ has no gap and it folds into $\mathcal{L}^{j+1}$. ■

Algorithm 6 describes the steps to compose two crease patterns $\mathcal{F}_1$ and $\mathcal{F}_2$ into a new crease pattern $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$. The new distal marker $\boldsymbol{p}'_d$ is the sum of the two inputs modulo $n_s l_s$ due to the wrap-around property of a tubular sheet.

B. Link Design as a Dubins Path Problem

We are now prepared to address the problem of constructing a link that connects two modules (usually joints) with desired poses in $\mathbb{R}^3$. In other words, given the proximal and distal frames $\mathcal{O}_p$ and $\mathcal{O}_d$ in $\mathbb{R}^3$, find the link $\mathcal{L}$ that produces the corresponding transformation between these frames. This section shows that this design problem is related to the Dubins path planning problem.

Recall from Section III that the *centerline* of an origami module is the curve connecting the base centers $\boldsymbol{o}_p$ and $\boldsymbol{o}_d$ along a path that is equidistant to all facets, except for the elbow fitting, where it is the circular arc centered at $\boldsymbol{o}_{icr}$ with endpoints $\boldsymbol{o}_p$ and $\boldsymbol{o}_d$. In other words, it is a circular arc of radius r

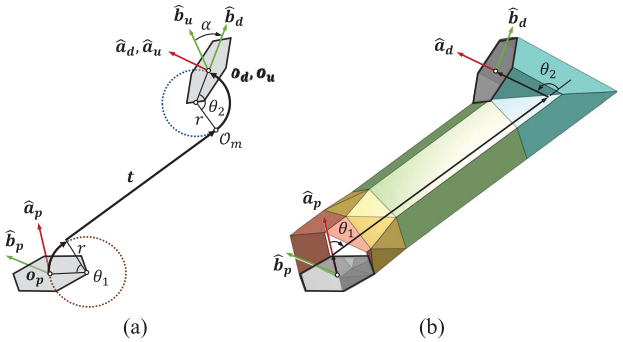


Fig. 5. Dubins-specified tube connecting method. (a) 3-D Dubins path planning. We want to transform the polygon at $\mathcal{O}_p$ to $\mathcal{O}_d$. The transformation of the polygon from $\mathcal{O}_p$ to $\mathcal{O}_d$ can be realized by finding a Dubins path that first takes the normal vector $\hat{a}$ of the polygon from $\mathcal{O}_p$ to $\mathcal{O}_d$ and then twists the frame for an angle α with respect to $\hat{a}_d$. (b) Dubins-specified tube connection is a sequence of an elbow fitting, followed by a twist fitting, a tube, and then another elbow fitting. All the parameters needed for this construction can be found with the 3-D Dubins path planning.

(denoted by C) for an elbow fitting and a line segment (denoted by S) for every other origami module (and any origami joint in its zero configuration), shown as the blue arrows in Fig. 3. We denote the centerline of the module $\mathcal{M}_i$ as $c(\mathcal{M}_i) \subset \mathbb{R}^3$. Similar to (33) and (35), centerlines can be transformed and composed in correspondence with the composition of their modules. In particular, the composition of two centerlines is $c(\mathcal{M}_1) \uplus c(\mathcal{M}_2) := c(\mathcal{M}_1) \cup ({}^{1p}\boldsymbol{M}_{1d} c(\mathcal{M}_2))$. It follows that the centerline of the link can be composed of the centerlines of individual modules.

Definition 2 (Centerline of a link): The centerline of a link $\mathcal{L}^1$ is denoted as $c(\mathcal{L}^1) = c(\mathcal{M}_1)$. The centerline of a link $\mathcal{L}^k$ is the concatenation of the centerlines of $\mathcal{L}^{k-1}$ and $\mathcal{M}_k$, or

$$c\left(\mathcal{L}^k\right) = c\left(\mathcal{L}^{k-1} \uplus \mathcal{M}_k\right) := c\left(\mathcal{L}^{k-1}\right) \uplus c\left(\mathcal{M}_k\right). \tag{46}$$

Observe that in the absence of twist, the centerline and base of a module (and thus of a link) are sufficient to identify the module and its design parameters uniquely. This work shows that any link can be constructed as the realization of an appropriate path from $\mathcal{O}_p$ to $\mathcal{O}_d$ consisting of C and S.

Specifically, we focus on a subset of paths called CSC paths, or the bounded-curvature paths commonly computed for Dubins vehicles [51]. Consider a generic CSC path that connects frames $\mathcal{O}_p$ and $\mathcal{O}_d$, where we follow a circular arc of radius r starting at $\boldsymbol{o}_p$ tangent to $\hat{\boldsymbol{a}}_p$ for a distance θ_1, then follow a straight line segment for a distance $||\mathbf{t}||$ until we reach the last circular arc of radius r, then follow that arc for another θ_2 until we finish at $\boldsymbol{o}_d$ tangent to $\hat{\boldsymbol{a}}_d$, as shown in Fig. 5(a). The rigid transformation that takes the vector $\hat{\boldsymbol{a}}_p$ at $\boldsymbol{o}_p$ to $\hat{\boldsymbol{a}}_d$ at $\boldsymbol{o}_d$ can be computed directly from this path. However, this transformation does not constrain the orientation of the frame, and an additional twist transformation is needed to fully describe the rigid transformation from $\mathcal{O}_p$ to $\mathcal{O}_d$. Here, we introduce a new frame to represent the new target goal, the untwisted distal frame $\mathcal{O}_u = \{\hat{\boldsymbol{a}}_u, \hat{\boldsymbol{b}}_u, \hat{\boldsymbol{c}}_u, \boldsymbol{o}_u\}$, or the frame whose centerline axis and origin are identical to the distal frame ($\hat{\boldsymbol{a}}_u = \hat{\boldsymbol{a}}_d$ and $\boldsymbol{o}_u = \boldsymbol{o}_d$) but not the other two axes. Frames $\mathcal{O}_u$ and $\mathcal{O}_d$ have an angular difference of α about the $\hat{\boldsymbol{a}}_d$ axis. Therefore

Lemma 3 (Rigid transformation of a CSC Dubins path with twist correction): Suppose a CSC Dubins path with turning radius r staring at $\boldsymbol{o}_p$ tangent to $\hat{\boldsymbol{a}}_p$ and finishing at $\boldsymbol{o}_d$ tangent to $\hat{\boldsymbol{a}}_d$ exists, with a twist α along the centerline. Then the rigid transformation from the associated proximal frame $\mathcal{O}_p$ to the distal frame $\mathcal{O}_d$ can be written as

$$\begin{aligned}\bar{\mathcal{O}}_d := &\begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) & d_{w1}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \\ &\times \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_a, \alpha) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & ||\boldsymbol{t}||\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \\ &\times \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w2'}, \theta_2) & d_{w2}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w2'}, \theta_2) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_p. \end{aligned} \tag{47}$$

Proof: We first consider the problem where there is no twist about $\hat{\boldsymbol{a}}_d$ (i.e., $\alpha = 0$), and the problem reduces simply to transforming the unit normal vectors $\hat{\boldsymbol{a}}_p$ at $\boldsymbol{o}_p$ to $\hat{\boldsymbol{a}}_d$ at $\boldsymbol{o}_d$. The homogeneous transformation from the proximal frame $\mathcal{O}_p$ to the untwisted distal frame $\mathcal{O}_u$ as shown in the CSC Dubins path from Fig. 5(a) can be factored into a sequence of three rigid transformations as follows. First, start with a turn of constant radius r over an arc of θ_1 radians. Second, follow a straight path of length $||\boldsymbol{t}||$ along the vector $\boldsymbol{t}$ that is tangential to both circular arcs of the CSC path. Finally, end with a turn of constant radius r over an arc of θ_2 radians. Algebraically, these three transformations correspond to the product

$$\begin{aligned}\bar{\mathcal{O}}_u := &\begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) & d_{w1}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & ||\boldsymbol{t}||\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \\ &\times \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w2}, \theta_2) & d_{w2}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w2}, \theta_2) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \mathbf{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_p \end{aligned} \tag{48}$$

where $\hat{\boldsymbol{e}}_a = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $\hat{\boldsymbol{e}}_{w1}$ is the rotational axis of the first circular arc in terms of $\mathcal{O}_p$, $\hat{\boldsymbol{e}}_{w2}$ is the rotational axis of the second circular arc in terms of $\mathcal{O}_m$ (the intermediate frame after the first circular arc and the straight line), $d_{w1} = r \tan \frac{\theta_1}{2}$, and $d_{w2} = r \tan \frac{\theta_2}{2}$.

To transform the untwisted frame $\mathcal{O}_u$ back to the distal frame $\mathcal{O}_d$, we perform a twist $\boldsymbol{R}(\hat{\boldsymbol{a}}_u, \alpha)$ on $\mathcal{O}_u$, or

$$\bar{\mathcal{O}}_d := \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_a, \alpha) & 0 \\ \mathbf{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_u. \tag{49}$$

Since the twist operation does not change the shape of the CSC centerline path, we can apply it after the first constant radius turn before the translation along the straight line component of the path. The transformation between $\mathcal{O}_p$ and $\mathcal{O}_d$ is then as given in (47). Since the twist transformation is before the last circular arc, the rotational axis must be written in the new frame accordingly, where $\hat{\boldsymbol{e}}_{w2'}$ is the unit rotational axis of the second circular arc observed in the body frame. ■

We can generalize the twist transformation into a screw transformation, having twist and translation simultaneously. Then,

we can rewrite (47) as

$$\bar{\mathcal{O}}_d := \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) & d_{w1}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w1}, \theta_1) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_a, \alpha) & \xi||\boldsymbol{t}||\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & (1-\xi)||\boldsymbol{t}||\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{R}(\hat{\boldsymbol{e}}_{w2'}, \theta_2) & d_{w2}(\boldsymbol{R}(\hat{\boldsymbol{e}}_{w2'}, \theta_2) + \boldsymbol{I})\hat{\boldsymbol{e}}_a \\ \boldsymbol{0} & 1 \end{bmatrix} \bar{\mathcal{O}}_p. \tag{50}$$

where $\xi \in (0, 1)$ is an arbitrary fraction. Essentially, we are splitting the line segment of the CSC path such that the first portion contains a screw operation (a twist $\boldsymbol{R}(\hat{\boldsymbol{e}}_a, \alpha)$ and a translation $h_t = \xi||\boldsymbol{t}||$) and the second portion is a translation operation (with $h = (1-\xi)||\boldsymbol{t}||$). Now observe that the four transformation matrices in (50) are the same as the transformation matrices of an elbow fitting (14), a twist fitting (10), a tube (7), and another elbow fitting (14) in sequential multiplication. Thus, we can construct a physical connection between two regular polygon bases by connecting the four origami modules in the mentioned order, where all the geometric parameters can be found in (50).

Corollary 4: Given (50), the origami link $\mathcal{L}^4 = \mathcal{M}_{ep} \uplus \mathcal{M}_{tw} \uplus \mathcal{M}_{tu} \uplus \mathcal{M}_{ed}$ (with polygon base of circumradius r) composed of elbow fittings $\mathcal{M}_{ep}$, $\mathcal{M}_{ed}$, a twist fitting $\mathcal{M}_{tw}$, and a tube $\mathcal{M}_{tu}$ instantiates the same given rigid transformation.

Proof: Choosing the base of the link to have a circumradius of r, the elbow fittings $\mathcal{M}_{ep}$ and $\mathcal{M}_{ed}$ have centerlines of circular arcs with radius r. The number of polygon sides n_s can be chosen arbitrarily (see Section VI-B). We now combine the elbow fitting $\mathcal{M}_{ep}$ (with rotation axis $\hat{\boldsymbol{e}}_{w1}$ and angle θ_1), twist fitting $\mathcal{M}_{tw}$ (with twist angle α and length $\xi||\boldsymbol{t}||$), tube $\mathcal{M}_{tu}$ (with length $(1-\xi||\boldsymbol{t}||)$), and elbow fitting $\mathcal{M}_{ed}$ (with rotation axis $\hat{\boldsymbol{e}}_{w2'}$ and angle θ_2) to form the link. From Lemma 1, we get the rigid transformation instantiated by $\mathcal{L}^4$ through multiplication of (7), (10), and (14), and again (14) with the aforementioned input parameters, which is then identical to (50). ■

In fact, the centerline of this constructed link is the CSC path provided in Lemma 3. We can now combine all the results and provide sufficiency conditions for the existence of a link that connects a proximal and distal frame $\mathcal{O}_p$ and $\mathcal{O}_d$ with a specified polygon base.

Lemma 5 (Dubins-specified origami link): Given a proximal frame $\mathcal{O}_p$, a distal frame $\mathcal{O}_d$, and a regular n_s-sided polygon of circumradius r, a link connecting frame $\mathcal{O}_p$ to $\mathcal{O}_d$ can be constructed if there exists a CSC Dubins path starting at $\boldsymbol{o}_p$ tangent to $\hat{\boldsymbol{a}}_p$ and finishing at $\boldsymbol{o}_d$ tangent to $\hat{\boldsymbol{a}}_d$. Furthermore, the link $\mathcal{L}^4 = \mathcal{M}_{ep} \uplus \mathcal{M}_{tw} \uplus \mathcal{M}_{tu} \uplus \mathcal{M}_{ed}$ (with regular n_s-sided polygon of circumradius r as its base) is composed of elbow fittings $\mathcal{M}_{ep}$, $\mathcal{M}_{ed}$, a twist fitting $\mathcal{M}_{tw}$, and a tube $\mathcal{M}_{tu}$, such that its centerline is the given Dubins path.

We refer to work in [51], which claims that such a CSC path can be computed as long as

$$\|\boldsymbol{o}_p - \boldsymbol{o}_d\| \geq 4r. \tag{51}$$

Algorithm 7: $DubinsLink(n_s, r, \mathcal{O}_p, \mathcal{O}_d)$.

Input: Polygon shape (n_s, r), the proximal frame and the distal frame $\mathcal{O}_p$, $\mathcal{O}_d$
Output: Crease pattern $\mathcal{G}$, distal marker $\boldsymbol{p}'_d$

$(\boldsymbol{t}, \theta_1, \theta_2) \leftarrow$ Find variables of a CSC path, e.g., [51]*;
$\hat{\boldsymbol{w}}_p \leftarrow \hat{\boldsymbol{a}}_p \times \hat{\boldsymbol{t}}$; $\hat{\boldsymbol{w}}_d \leftarrow \hat{\boldsymbol{t}} \times \hat{\boldsymbol{a}}_d$;
$\hat{\boldsymbol{b}}_m \leftarrow R(\hat{\boldsymbol{w}}_p, \theta_1)\hat{\boldsymbol{b}}_p$; $\hat{\boldsymbol{b}}_u \leftarrow R(\hat{\boldsymbol{w}}_d, \theta_2)\hat{\boldsymbol{b}}_m$;
$\alpha \leftarrow \text{atan2}\Big((\hat{\boldsymbol{b}}_u \times \hat{\boldsymbol{b}}_d) \cdot \hat{\boldsymbol{a}}_d, \hat{\boldsymbol{b}}_u \cdot \hat{\boldsymbol{b}}_d\Big)$;
$\phi_1 \leftarrow \text{atan2}\Big((\hat{\boldsymbol{b}}_p \times \hat{\boldsymbol{w}}_p) \cdot \hat{\boldsymbol{a}}_p, \hat{\boldsymbol{b}}_p \cdot \hat{\boldsymbol{w}}_p\Big)$;
$\phi_2 \leftarrow \text{atan2}\Big((\hat{\boldsymbol{b}}_m \times \hat{\boldsymbol{w}}_d) \cdot \hat{\boldsymbol{t}}, \hat{\boldsymbol{b}}_m \cdot \hat{\boldsymbol{w}}_d\Big) - \alpha$;
$(\mathcal{G}, \boldsymbol{p}'_d) \leftarrow ElbowFitting(n_s, r, \theta_1, \phi_1)$;
$(\mathcal{F}_2, \boldsymbol{p}'_{2d}) \leftarrow TwistFitting(n_s, r, \alpha, 0.2||\boldsymbol{t}||)$;
$(\mathcal{F}_3, \boldsymbol{p}'_{3d}) \leftarrow Tube(n_s, r, 0.8||\boldsymbol{t}||)$;
$(\mathcal{F}_4, \boldsymbol{p}'_{4d}) \leftarrow ElbowFitting(n_s, r, \theta_2, \phi_2)$;
for $i = 2, 3, 4$ **do**
  $(\mathcal{G}, \boldsymbol{p}'_d) \leftarrow ComposeCP\Big((\mathcal{G}, \boldsymbol{p}'_d), (\mathcal{F}_i, \boldsymbol{p}'_{id})\Big)$;
end

*Note the change in variables from X_1, X_2, X, v_1, v_2 in the original text [51] to $\boldsymbol{o}_p$, $\boldsymbol{o}_d$, $\boldsymbol{t}$, $\hat{\boldsymbol{a}}_p$, $\hat{\boldsymbol{a}}_d$ for our work.

Note that not all CSC Dubins paths can produce feasible Dubins-specified links. If the arc angle of either C portion is greater than $3\pi/2$, the generated link will self-intersect (we will discuss self-intersection avoidance in Section V-B). Algorithm 7 shows the entire algorithm for constructing a specified link. Without loss of generality, we chose $\xi = 0.2$ as an example for the rest of the study.

V. Joint Placement

Having just shown how to compose modules into specified links, we now address the problem of composing links into specified kinematic chains. Namely, this section considers the relaxed configuration ($\boldsymbol{q}_0 = \boldsymbol{0}$) and determines where to place joints in $\mathbb{R}^3$ such that their operation realizes the required forward kinematics specified by D-H parameters $\mathcal{D}$ when actualized by the tubular origami designs. Given an origami joint module $\mathcal{J}_i$ constructed in Section III-B, we define the joint centroid frame $\mathcal{O}_{ic} = \{\hat{\boldsymbol{a}}_{ic}, \hat{\boldsymbol{b}}_{ic}, \hat{\boldsymbol{c}}_{ic}, \boldsymbol{o}_{ic}\}$. The origin of this frame represents the midpoint between the proximal and the distal base origins $(\boldsymbol{o}_{ip}, \boldsymbol{o}_{id})$ in the zero configuration of $\mathcal{J}_i$, as shown in Fig. 6(a) and (b). For each $\mathcal{J}_i$, frames $\mathcal{O}_{ic}$, $\mathcal{O}_{ip}$, and $\mathcal{O}_{id}$ all have the same orientation, where $\hat{\boldsymbol{c}}_{ic}$ is defined as the rotation axis for a revolute joint and $\hat{\boldsymbol{a}}_{ic}$ as the translational axis for a prismatic joint. Assuming we have all the joint frames $\mathcal{O}_i$ (they are fixed frames when given $\mathcal{D}$ and its forward kinematics), we want to solve the following problem.

Problem 3: Given the joint frames $\mathcal{O}_i$, identify $\mathcal{O}_{ic}$ for all joints such that they satisfy the D-H specification (by sharing the same joint axis with $\mathcal{O}_i$), and a Dubins-specified origami link can be constructed between any two joints (by ensuring the joints are sufficiently distant).

For practical purposes, it is desirable to find joint locations that minimize the volume of the linkage. Thus, this problem can be formulated as a constrained packing problem. Note that the problem of finding the optimal placement of the joints

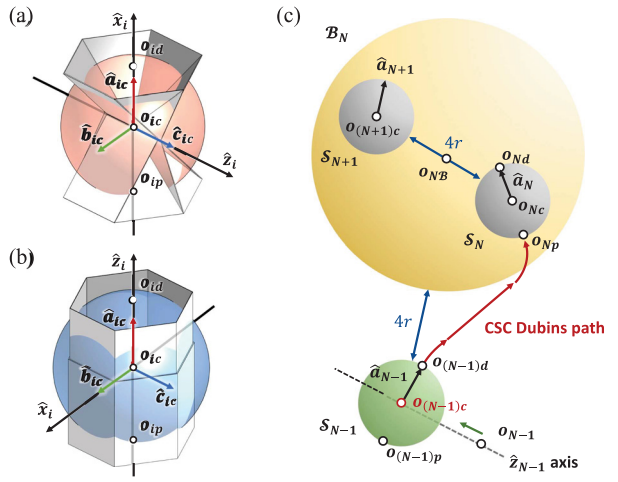


Fig. 6. Origami joint placement: The $\mathcal{O}_{ic}$ frame and the joint sphere $\mathcal{S}_i$ of the (a) revolute and (b) prismatic joints. (c) Joint placement for $\mathcal{J}_{N-1}$. Section V proposes a method to ensure each joint is at least a distance d apart from any other. Each joint is represented as a sphere and can be placed anywhere on its $\hat{\boldsymbol{z}}_i$ axis. $\mathcal{B}_N$ is the minimum bounding sphere for $\mathcal{S}_N \cup \mathcal{S}_{N+1}$. The red path is the CSC Dubins path that will connect $\mathcal{J}_{N-1}$ and $\mathcal{J}_N$.

Algorithm 8: $JointPlacement(\mathcal{D}, n_s, r, \boldsymbol{q}_0, \boldsymbol{q}_m)$.

Input: D-H parameter table $\mathcal{D}$, polygon shape (n_s, r), the maximum revolute joint range $\boldsymbol{q}_m$, and the zero configuration $\boldsymbol{q}_0$

Output: List of origami joint centroid frame $\boldsymbol{\mathcal{O}}_c$

Find the joint frames $\mathcal{O}_i$ with forward kinematics assuming $q_i = 0$ for all i;
$\mathcal{S}_{N+1} \leftarrow$ ball $(\boldsymbol{o}_{N+1}, r)$; $\mathcal{B}_{N+1} \leftarrow \mathcal{S}_{N+1}$;
$\mathcal{O}_{(N+1)c} \leftarrow$ elements from $\mathcal{O}_{N+1}$ such that $\hat{\boldsymbol{a}}_{(N+1)c}$ represents the end effector approach direction;
for $i = N$ **to** 1 **do**
  $r_{is} \leftarrow$ assign with Eq. (53);
  $\mathcal{O}_{ic} \leftarrow$ assign Eq. (52) with (t_i, u_i) such that
    $t_i \leftarrow t^*$ that minimizes $(\boldsymbol{o}_i + t^* \hat{\boldsymbol{z}}_i) - \boldsymbol{o}_{(i+1)c}$ s.t. dist $(\boldsymbol{o}_{(i+1)\mathcal{B}}, \boldsymbol{o}_i + t^* \hat{\boldsymbol{z}}_i) \geq r_{(i+1)\mathcal{B}} + 4r + r_{is}$;
    $u_i \leftarrow 1$ or -1 s.t. $\hat{\boldsymbol{a}}_{ic}{}^T$ normal $(\boldsymbol{o}_{ic}, \boldsymbol{o}_{(i+1)c}) \geq 0$;
  $\mathcal{S}_i \leftarrow$ ball$(\boldsymbol{o}_{ic}, r_{is})$;
  $\mathcal{B}_i \leftarrow$ ball$(\boldsymbol{o}_{i\mathcal{B}}, r_{i\mathcal{B}})$ that encloses $\mathcal{S}_i \cup \mathcal{B}_{i+1}$;
end
$\boldsymbol{\mathcal{O}}_c \leftarrow \{\mathcal{O}_{1c}, \mathcal{O}_{2c}, ..., \mathcal{O}_{Nc}, \mathcal{O}_{(N+1)c}\}$.

(i.e., such that the distance between joint pairs is minimized) is NP-hard [56]. However, a feasible nonoptimal solution can be obtained through an iterative greedy approach.

Given a D-H specification $\mathcal{D}$, the transformation matrix ${}^{i-1}\boldsymbol{T}_i$ that specifies the position and orientation of each joint frame $\mathcal{O}_i := \{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i, \boldsymbol{o}_i\}$ can be computed directly [47]. Thus, the forward kinematics for the target linkage is fully specified. Note, however, that although the frame and the kinematic motion for each joint are defined through $\mathcal{D}$, the exact joint locations $\mathcal{O}_{ic}$ are not. A revolute joint $\mathcal{J}_i$, for example, can be placed anywhere as long as its rotational axis $\hat{\boldsymbol{c}}_{ic}$ is aligned with $\hat{\boldsymbol{z}}_i$; thus we can assign $\hat{\boldsymbol{c}}_{ic} := \hat{\boldsymbol{z}}_i$ and $\boldsymbol{o}_{ic} := \boldsymbol{o}_i + t\hat{\boldsymbol{z}}_i$ for any $t \in \mathbb{R}$. On the other hand, a prismatic joint $\mathcal{J}_j$ can be placed anywhere as long as its translational axis $\hat{\boldsymbol{a}}_{ic}$ is aligned with $\hat{\boldsymbol{z}}_j$; thus we can assign $\hat{\boldsymbol{a}}_{ic} := \hat{\boldsymbol{z}}_i$ and $\boldsymbol{o}_{ic} := \boldsymbol{o}_i + t\hat{\boldsymbol{z}}_i$ for any $t \in \mathbb{R}$. The remaining two axes of $\mathcal{O}_{ic}$ do not affect the kinematics of the mechanism and can be chosen arbitrarily. In this work, we assign $\hat{\boldsymbol{a}}_{ic} := \hat{\boldsymbol{x}}_i$ for an origami revolute joint and $\hat{\boldsymbol{b}}_{ic} := \hat{\boldsymbol{x}}_i$ for an origami prismatic joint; the last axis follows the right-hand rule. Further, due to the rotational symmetry of the origami joint, we can rotate the frame about $\hat{\boldsymbol{c}}_{ic}$ by π without changing the kinematics. Due to this flexibility in precise joint location, we can, therefore, assign $\mathcal{O}_{ic} = \{\hat{\boldsymbol{a}}_{ic}, \hat{\boldsymbol{b}}_{ic}, \hat{\boldsymbol{c}}_{ic}, \boldsymbol{o}_{ic}\}$ as

$$\begin{aligned} \mathcal{O}_{ic} &:= \{u_i\hat{\boldsymbol{z}}_i, u_i\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i, \boldsymbol{o}_i + t_i\hat{\boldsymbol{z}}_i\} \text{ for a prismatic joint} \\ \mathcal{O}_{ic} &:= \{u_i\hat{\boldsymbol{x}}_i, u_i\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i, \boldsymbol{o}_i + t_i\hat{\boldsymbol{z}}_i\} \text{ for a revolute joint} \end{aligned} \quad (52)$$

where the design parameter $t_i \in \mathbb{R}$ determines the joint centroid location and the choice of the scalar $u_i = \pm 1$ dictates whether the frame has been rotated. In practice, we choose the direction of $\hat{\boldsymbol{a}}_{ic}$ to point toward the next joint for a shorter centerline path, thus reducing the sheet material needed.

A. General Joint Placement

Problem 3 thus reduces to identifying for each joint $\mathcal{J}_i$ the offset t_i required to guarantee that a connecting link can be constructed. We propose a greedy iterative approach to solving this problem (Algorithm 8), wherein joint locations are determined iteratively such that each new joint location is a distance of at least $4r$ away from all the previous joints, spreading the joints out to make room for a Dubins-specified link (51), as illustrated in Fig. 6(b). Since this procedure creates longer (hence, heavier) links in later iterations, we place the joints in reverse order such that longer links are closer to the base.

Denote by ball$(\boldsymbol{o}, r)$ the set of points at a distance of at most r from $\boldsymbol{o}$. We define a *joint sphere* $\mathcal{S}_i := \text{ball}(\boldsymbol{o}_{ic}, r_{is})$ for an origami joint $\mathcal{J}_i$, shown in Fig. 6(a), where its radius is chosen such that the proximal and distal origins both lie on the sphere surface, i.e.,

$$\begin{aligned} r_s &:= r \sin \tfrac{(n_s-2)\pi}{2n_s} \tan \tfrac{\theta_m}{4} \text{ for a revolute joint} \\ r_s &:= \tfrac{1}{4} d_0 (2 + \csc \beta) \text{ for a prismatic joint.} \end{aligned} \quad (53)$$

Denote by $\mathcal{B}_j := \text{ball}(\boldsymbol{o}_{j\mathcal{B}}, r_{j\mathcal{B}})$ the minimum bounding sphere that encloses all the joint spheres from $\mathcal{S}_j$ to $\mathcal{S}_{N+1}$. For two compact sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^3$, denote by dist$(\mathcal{S}_1, \mathcal{S}_2)$ the shortest distance between $\mathcal{S}_1$ and $\mathcal{S}_2$, and normal$(\mathcal{S}_1, \mathcal{S}_2)$ the vector from a point in $\mathcal{S}_1$ to a point in $\mathcal{S}_2$ such that the length is dist$(\mathcal{S}_1, \mathcal{S}_2)$. Denote by line$(\boldsymbol{o}, n)$ the line that passes through the point $\boldsymbol{o}$ in the direction of $\hat{\boldsymbol{n}}$.

Lemma 6: Algorithm 8 solves Problem 3.

Proof: The proof must show that for all $N \in \mathbb{N}$, Algorithm 8 accepts $\mathcal{D}$, a D-H list of length $N+1$, and returns a list of joint centroid frames $\boldsymbol{\mathcal{O}}_c$ that realize the kinematic mapping (1) specified by $\mathcal{D}$ at rest configuration q_0, and that each pair of consecutive joint centroid frames is far enough for a Dubins-specified link to exist between them. We will prove this by

induction on N. For the base case, $N = 1$, there are two entries in $\mathcal{D}$, the final end effector in row $N + 1 = 2$ and the single joint in row $N + 0 = 1$. In this case, Algorithm 8 uses (53) to place $\mathcal{O}_{2c}$ at the center of ball$(\boldsymbol{o}_2, r_{2\,s})$, which serves as both the joint sphere, $\mathcal{S}_2$, and the ball, $\mathcal{B}_2$. Algorithm 8 then assigns the joint centroid frame, $\mathcal{O}_{1c}$, at a distance $4r + r_{1\,s}$ from $\mathcal{B}_2$ along the specified line line$(\boldsymbol{o}_1, \hat{\boldsymbol{z}}_1)$, and then, updates $\mathcal{B}_1$ to enclose $\mathcal{S}_1 \cup \mathcal{B}_2$ as depicted in Fig. 6(c). For the inductive step, assume that the proposition holds for any specification, $\mathcal{D}$, of length N. Given a specification $\tilde{\mathcal{D}}$ of length $N + 1$, strip its first row (which we number row 1) and apply Algorithm 8 to $\mathcal{D}$, its subsequent N rows. Once again, apply Algorithm 8 to adjoin a joint centroid frame, $\mathcal{O}_{1c}$, at a distance $4r + r_{1\,s}$ from the final ball, $\mathcal{B}_2$ associated with $\mathcal{D}$, along the specified line line$(\boldsymbol{o}_0, \hat{\boldsymbol{z}}_0)$, and then update $\mathcal{B}_1$ to enclose $\mathcal{S}_1 \cup \mathcal{B}_2$, completing the list $\boldsymbol{\mathcal{O}}_c$ as required. ■

B. Avoiding Self-Intersection

Algorithm 8 only ensures that joints are placed in locations such that CSC paths can be constructed between them. There is no guarantee that the resulting links do not intersect with themselves or each other. To start, if an elbow fitting has a bending angle greater than π, the two links will intersect. We conjecture that it is possible to construct a CSC path such that both the arc angles are less than π. Denote by $\mathcal{P} := \text{plane}(\boldsymbol{o}, \hat{\boldsymbol{n}})$ the plane that passes through the point $\boldsymbol{o}$ with the normal vector $\hat{\boldsymbol{n}}$ and denote by $\mathcal{P}^+$ the half-space divided by $\mathcal{P}$ that contains all nonnegative values.

Conjecture 7 (Constrained CSC Dubins path): Given a proximal frame $\mathcal{O}_p$ and a distal frame $\mathcal{O}_d$, if there exist two parallel planes $\mathcal{P}_p := \text{plane}(\boldsymbol{o}_p \hat{\boldsymbol{n}})$ and $\mathcal{P}_d := \text{plane}(\boldsymbol{o}_d, \hat{\boldsymbol{n}})$, such that (1) $\text{dist}(\mathcal{P}_p, \mathcal{P}_d) \geq 4r$, (2) $\hat{\boldsymbol{n}}^T \hat{\boldsymbol{a}}_p \geq 0$, and (3) $\hat{\boldsymbol{n}}^T \hat{\boldsymbol{a}}_d \geq 0$, then there exists a shortest CSC path of radius r starting at $\boldsymbol{o}_p$ tangent to $\hat{\boldsymbol{a}}_p$ and finishing at $\boldsymbol{o}_d$ tangent to $\hat{\boldsymbol{a}}_d$ whose circular arcs both subtend angles less than π radians.

Assuming the conjecture is true, we now introduce a procedure to spread out the joint spheres so that no two links can cross. This is done by introducing *waypoints* to reroute the centerline of the link to avoid path intersections. A waypoint $\mathcal{W}$ has a frame $\mathcal{O}_{\mathcal{W}}$ and can be treated as a particular joint that allows no motion and has no volume. Its joint sphere is a point (radius $r_s = 0$) and its crease pattern can be constructed with $tube(n_s, r, 0)$. We propose Algorithm 9 to identify $\mathcal{O}_{ic}$ for every joint and add waypoints to reroute the centerline of a link such that the constructed link that connects any two joints does not intersect with itself or any other links.

Algorithm 9: $JointPlacement2(\mathcal{D}, n_s, r, \boldsymbol{q}_0, \boldsymbol{q}_m)$.

Input: D-H parameter table $\mathcal{D}$, polygon shape (n_s, r), the maximum revolute joint range $\boldsymbol{q}_m$, and the zero configuration $\boldsymbol{q}_0$

Output: list of origami joint centroid frame $\boldsymbol{\mathcal{O}}_c$

1 Find the joint frames $\mathcal{O}_i$ with forward kinematics assuming $q_i = 0$ for all i;
2 $\mathcal{S}_{N+1} \leftarrow \text{ball}(\boldsymbol{o}_{N+1}, r)$; $\mathcal{B}_{N+1} \leftarrow \mathcal{S}_{N+1}$;
3 $\mathcal{O}_{(N+1)c} \leftarrow$ elements from $\mathcal{O}_{N+1}$ such that $\hat{\boldsymbol{a}}_{(N+1)c}$ represents the end effector approach direction;
4 **for** $i = N$ **to** 1 **do**
5 $r_{is} \leftarrow$ assign with Eq. (53);
6 $\hat{\boldsymbol{n}}_{i1} \leftarrow -\hat{\boldsymbol{a}}_{jc}$; /* denote j:=i+1 */
7 $\mathcal{P}_{i1} \leftarrow \text{plane}(o_{j\mathcal{B}} + r_{j\mathcal{B}} \hat{\boldsymbol{n}}_{i1}, \hat{\boldsymbol{n}}_{i1})$;
8 $\mathcal{P}'_{i1} \leftarrow \text{plane}\left(o_{j\mathcal{B}} + (r_{j\mathcal{B}} + 4r + r_{is}) \hat{\boldsymbol{n}}_{i1}, \hat{\boldsymbol{n}}_{i1}\right)$;
9 $\boldsymbol{o}_{i\mathcal{W}1} \leftarrow \text{line}(\boldsymbol{o}_{jc}, \hat{\boldsymbol{n}}_{i1}) \cap \mathcal{P}_{i1}$;
10 $\mathcal{O}_{i\mathcal{W}1} \leftarrow \{\hat{\boldsymbol{a}}_{jc}, \hat{\boldsymbol{b}}_{jc}, \hat{\boldsymbol{c}}_{jc}, \boldsymbol{o}_{i\mathcal{W}1}\}$;
11 **if** $\text{line}(\boldsymbol{o}_i, \hat{\boldsymbol{z}}_i) \cap \mathcal{P}'^{+}_{i1} = \varnothing$ **then**
12 $\hat{\boldsymbol{n}}_{i2} \leftarrow -\hat{\boldsymbol{z}}_i$;
13 $\mathcal{P}_{i2} \leftarrow \text{plane}(o_{j\mathcal{B}} + r_{j\mathcal{B}} \hat{\boldsymbol{n}}_{i2}, \hat{\boldsymbol{n}}_{i2})$;
14 $\mathcal{P}'_{i2} \leftarrow \text{plane}(o_{j\mathcal{B}} + (r_{j\mathcal{B}} + 4r + r_{is}) \hat{\boldsymbol{n}}_{i2}, \hat{\boldsymbol{n}}_{i2})$;
15 $\boldsymbol{o}_{i\mathcal{W}2} \leftarrow \text{line}(\boldsymbol{o}_{i\mathcal{W}1} + r \hat{\boldsymbol{n}}_{i1}, \hat{\boldsymbol{n}}_{i2}) \cap \mathcal{P}_{i2}$;
16 $\mathcal{O}_{i\mathcal{W}2} \leftarrow \{\boldsymbol{R}(\hat{\boldsymbol{n}}_{i1} \times \hat{\boldsymbol{n}}_{i2}, \frac{\pi}{2})[\,\hat{\boldsymbol{a}}_{jc}\ \hat{\boldsymbol{b}}_{jc}\ \hat{\boldsymbol{c}}_{jc}\,], \boldsymbol{o}_{i\mathcal{W}2}\}$;
17 **else**
18 $(\hat{\boldsymbol{n}}_{i2}, P_{i2}, P'_{i2}, \mathcal{O}_{i\mathcal{W}2}) \leftarrow (\hat{\boldsymbol{n}}_{i1}, P_{i1}, P'_{i1}, \mathcal{O}_{i\mathcal{W}1})$;
19 **end**
20 $\mathcal{O}_{ic} \leftarrow$ assign Eq. (52) with (t_i, u_i) such that
$t_i \leftarrow t^*$ that minimises $(\boldsymbol{o}_i + t^* \hat{\boldsymbol{z}}_i) - \boldsymbol{o}_{i\mathcal{W}2}$
subject to $(\boldsymbol{o}_i + t^* \hat{\boldsymbol{z}}_i) \subset \mathcal{P}'^{+}_{i2}$;
$u_I \leftarrow 1$ or -1 such that $-\hat{\boldsymbol{a}}_{ic}{}^T \boldsymbol{n}_{i2} \geq 0$;
21 $\boldsymbol{\mathcal{O}}_{ic} \leftarrow \{\mathcal{O}_{ic}, \mathcal{O}_{i\mathcal{W}2}, \mathcal{O}_{i\mathcal{W}1}, \boldsymbol{\mathcal{O}}_{jc}\}$;
22 $\mathcal{S}_i \leftarrow \text{ball}(\boldsymbol{o}_{ic}, r_{is})$;
23 $\mathcal{B}_i \leftarrow \text{ball}(\boldsymbol{o}_{i\mathcal{B}}, r_{i\mathcal{B}})$ that encloses $\mathcal{S}_i \cup \mathcal{B}_j$;
24 **end**

VI. Origami Kinematic Chain

With the origami links $\mathcal{L}$ and joints $\mathcal{J}$ defined, we now define the origami kinematic chain recursively as

Definition 3 (Origami kinematic chain): A Kinegami chain is the composition of $N + 1$ links and N joints of the same polygon base in alternating order.

In the case of $N = 0$, we define the base link $\mathcal{L}_0$, although it cannot move, as an instance of an origami kinematic chain $\mathcal{K}_0$. A kinematic chain $\mathcal{K}_k$ with k joints can then be formed through the composition of the previous kinematic chain $\mathcal{K}_{k-1}$, an origami joint $\mathcal{J}_k$, and a link $\mathcal{L}_k$. In other words

$$\mathcal{K}_0 := \mathcal{L}_0, \text{ and } \mathcal{K}_k := \mathcal{K}_{k-1} \uplus \mathcal{J}_k \uplus \mathcal{L}_k. \tag{54}$$

The crease pattern of a kinematic chain is constructed through the composition of the crease patterns of the corresponding links and joints (Lemma 2). Finally, we combine all previous results and algorithms to produce Algorithm 10, or the "Kinegami" algorithm, for generating a kinematic chain mechanism. Note that the number of the polygon sides n_s must be even if the proposed kinematic chain consists of revolute joints (see Section III-B2). Now we present our main result.

Theorem 8 (Kinegami): Algorithm 10 solves Problem 1.

Proof: We prove this by induction. For the base step, line 1 generates the crease pattern $\mathcal{G}_0$ that folds into a tube of length l_b, which is the base case of a kinematic chain $\mathcal{K}_0$. For the inductive step, assume $\mathcal{G}_{j-1}$ folds into a kinematic chain $\mathcal{K}_{j-1}$. Since the joint location of $\mathcal{J}_j$ is assigned with Algorithm 8, it satisfies the constraints in Problem 3. According to Lemma 5, we can then generate a Dubins-specified link $\mathcal{L}_j$ that connects the consecutive pair of joints $\mathcal{J}_{j-1}, \mathcal{J}_j$. By composing $\mathcal{K}_{j-1}$, $\mathcal{J}_j$, and $\mathcal{L}_j$, we have the kinematic chain $\mathcal{K}_j$. In the special case

Algorithm 10: $Kinegami(\mathcal{D}, n_s, r, \boldsymbol{q}_0, \boldsymbol{q}_m)$.

Input: D-H parameter table $\mathcal{D}$, polygon shape (n_s, r), the maximum revolute joint range $\boldsymbol{q}_m$, and the zero configuration $\boldsymbol{q}_0$

Output: Crease pattern $\mathcal{G}$

1 $(\mathcal{G}_0, \boldsymbol{p}'_d) \leftarrow Tube(n_s, r, l_b)$;
2 $\boldsymbol{\mathcal{O}}_c \leftarrow JointPlacement(\mathcal{D}, n_s, r, \boldsymbol{q}_0, \boldsymbol{q}_m)$;
3 find $\boldsymbol{\mathcal{O}}_p$ and $\boldsymbol{\mathcal{O}}_d$ given $\boldsymbol{\mathcal{O}}_c$ and r_{is};
4 **for** $i = 1$ **to** $N + 1$ **do**
5 $(F_i, \boldsymbol{p}'_{id}) \leftarrow$ crease pattern according to joint type: $RevoluteJoint(n_s, r, \theta_m, n_z)$ for an R joint, $PrismaticJoint(n_s, r, d_0, n_l, \beta)$ for a P joint, $Tube(n_s, r, 0)$ for a waypoint;
6 $(\mathcal{G}_i, \boldsymbol{p}'_d) \leftarrow ComposeCP\left((\mathcal{G}_{i-1}, \boldsymbol{p}'_d), (\mathcal{F}_i, \boldsymbol{p}'_{id})\right)$;
7 $(\mathcal{G}_{i\mathcal{L}}, \boldsymbol{p}'_{i\mathcal{L}d}) \leftarrow DubinsLink(n_s, r, \mathcal{O}_{id}, \mathcal{O}_{(i+1)p})$;
8 $(\mathcal{G}_i, \boldsymbol{p}'_d) \leftarrow ComposeCP\left((\mathcal{G}_i, \boldsymbol{p}'_d), (\mathcal{G}_{i\mathcal{L}}, \boldsymbol{p}'_{i\mathcal{L}d})\right)$;
9 **end**

of the joint being a waypoint (see Section V-B), the algorithm generates a tube with no length, essentially composing the new link onto the previous kinematic chain. The crease pattern of $\mathcal{J}_j$ and $\mathcal{L}_j$ is $\mathcal{F}_j$ and $\mathcal{G}_{j\mathcal{L}}$, respectively. Based on Lemma 2, crease pattern $\mathcal{G}_j = \mathcal{G}_{j-1} \uplus \mathcal{F}_j \uplus \mathcal{G}_{j\mathcal{L}}$ folds into $\mathcal{K}_j$. ■

Lemma 9: Algorithm 10 outputs a crease pattern with $O(N)$ vertices and edges.

Proof: For rigid connections, Algorithms 1–3 produce crease patterns for origami modules with $O(n_s)$ vertices and edges. Each Dubins-specified link consists of up to four origami modules, and thus, Algorithm 7 also produces a crease pattern with $O(n_s)$ vertices and edges. For joints, Algorithm 4 produces a prismatic joint with $O(n_s)$ vertices and edges. The parameter n_l is a variable that can be chosen arbitrarily by the designer and does not need to scale with the problem. Algorithm 5 produces a revolute joint with $O(n_s n_z)$ vertices and edges. The recursive sink gadget adds additional vertices and edges for every n_z layer in every n_s iteration. Thus, the size of each module is at most $O(n_s n_z)$ vertices and edges.

Algorithm 10 connects N pairs of joints and Dubins-specified links and thus produces at most $O(N n_s n_z)$ vertices and edges. Note that n_s is a design variable and does not need to scale with the proposed problem. Equation (32) shows that the number of layers of the recursive sink gadget in the revolute joint is approximately proportional to its effective stiffness (i.e., $k \propto n_z$), and Algorithm 10 produces a crease pattern with $O(Nk)$ vertices and edges. For applications without stiffness constraints, we have $O(N)$ vertices and edges. ■

Lemma 10: Algorithm 10 runs in $O(N)$ time.

Proof: Calculating joint placement in Algorithm 8 requires solving a nonlinear program in line 6 and 8 for each joint. The scale of the equations does not change with the complexity of the serial manipulator and solves in $O(1)$. Since the process is repeated for all N joints, this algorithm takes $O(N)$ time. All modules are constructed vertex by vertex directly from the input parameters, and thus the time to build any individual module is $O(n_s n_z)$. Computing a Dubins-specified link requires solving a set of nonlinear equations in time independent of the complexity of the serial manipulator and thus takes $O(1)$. Composing crease patterns with Algorithm 6 takes $O(1)$, since shifting a graph in $\mathbb{R}^2$ is an addition operation. This process is repeated for all $N + 1$ links, so the entire process takes at most $O(N n_s n_z)$ time. Since n_s is a design parameter, the time complexity of Algorithm 10 is $O(N)$. ■

A. Programmable Compliance

We demonstrated that our revolute and prismatic joints have programmable stiffness in Section III-B. Although the precise relationship between parameters and stiffness has not yet been fully characterized, we observe that the stiffness of the revolute joint was approximately proportional to the number of sink folds n_z and the cone angle β. Denote k the desired stiffness, E the Young's modulus of the given material, and ϵ the error allowance. Assume there exist some functions f and g that map k, E, and ϵ to the geometric parameters n_z for the revolute joint and β for the prismatic joint, respectively. We then introduce this conjecture to be explored in future work:

Conjecture 11: With additional functions $n_z := f(k, \epsilon, E)$ and $\beta := g(k, \epsilon, E)$, Algorithm 10 solves Problem 2, where we can construct the compliant origami robot such that the local stiffness profile is bounded within $\boldsymbol{k} \pm \epsilon$. The pattern requires $O(Nk)$ time to compute and has $O(Nk)$ vertices and edges.

B. Selection of Polygon Shape

In general, the shape of the regular polygon base is a free design choice. Appendix A shows that for a fixed circumference tubular sheet, P_T, the second moment of the associated regular polygon prism tube—and hence, its ability to withstand bending—grows with n_s. However, crease pattern complexity increases with n_s (Lemmas 9 and 10), lengthening fabrication time respecting both computational and human folding effort. Finally, r directly affects the length scale and physical volume of the folded Kinegami output, particularly after imposing the sufficient conservative condition guaranteeing the existence of the Dubins-specified link.

VII. Physical Examples

The proposed "Kinegami" algorithm accepts an algebraic specification (D-H description) of a serial robot and outputs a single-sheet crease pattern that can be folded into an origami kinematic chain that is kinematically equivalent to the specification. The algorithm was implemented in MATLAB and can be found in our GitHub repository (https://github.com/weinitor/Kinegami). This section illustrates the use of Kinegami by physically constructing several common kinematic specifications. Specifically, in Section VII-A, we automatically generate and manually construct Kinegami implementations of some familiar robot arms that incorporate higher DOF lower pair joints. In Section VII-B, we illustrate the complexity challenge of this problem domain by exhibiting a faulty (self-intersecting) Kinegami chain produced by the naive link generator (Algorithm 8). We replace it with a feasible but conservatively bulky crease pattern

automatically produced by our conjecturally correct link generator (Algorithm 9). Entrusting the optimization step to human intervention instead, we suggest the ergonomic efficacy of the prior pipeline steps by exhibiting a feasible, correct, and more usefully compact design resulting from manually replacing in Algorithm 10 the call to Algorithm 8 (on line 2) with an intuitively generated selection of the joint centroid frames. Finally, we suggest the long-term value of origami robotics for exploiting cascaded power trains [36] by automatically generating, manually constructing, and empirically testing a 1-DOF Kinegami catapult. All the resulting origami robots are constructed out of 8-mil thick Durilla synthetics paper with polyester finish (CTI Paper, USA). The crease patterns are cut out and the folds perforated using a laser cutter. The sheet is then manually folded into shape and glued together with 3 M 467MP adhesive transfer tape. The manual folding time is roughly about $N \times 10$ min.

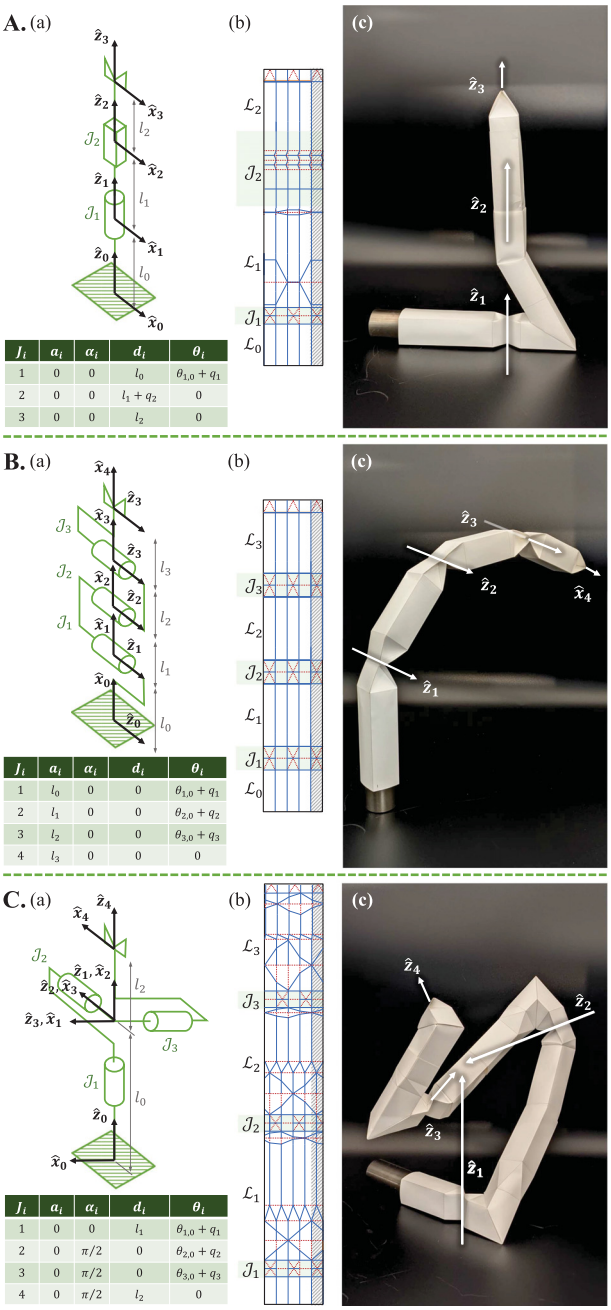
A. Higher DOF Lower Pair Joints

We construct origami serial robots that are kinematically equivalent to higher DOF lower pair joints. Let the polygon shape of the tubular origami be $\{n_s, r\} = \{4, 0.02\text{ m}\}$ and the maximum rotation angle for all the revolute joints be $\theta_m = \pi$.

1) Cylindrical Manipulator: A *cylindrical joint* permits sliding parallel to and rotation about its joint axis. Our cylindrical arm robot consists of a revolute and a prismatic joint, where both the joint axes coincide. Fig. 7(A)-(a) summarizes its kinematic diagram and D-H specification, where $l_0 = l_2 = 0.1$ m, $l_1 = 0.08$ m, and $\theta_{1,0} = 0$. Fig. 7(A)-(b) shows its crease pattern, where the ith joint $\mathcal{J}_i$ is highlighted in green and the ith link $\mathcal{L}_i$ uncolored. Fig. 7(A)-(c) shows the final product, where the ith joint axis $\hat{\boldsymbol{z}}_i$ is marked to show that the folded Kinegami chain is kinematically equivalent to the proposed robot.

2) Planar Manipulator: A *planar joint* permits arbitrary translation on and rotation perpendicular to a plane. Our 3-DOF manipulator arm is constructed with three parallel revolute joints, shown in Fig. 7(B), where $l_0 = l_1 = l_2 = l_3 = 0.15$ m and $\theta_{1,0} = \theta_{2,0} = \theta_{3,0} = 0$. Here, we choose $\theta_m = 3\pi/2$ to show a different range of motions for the revolute joint.

3) Spherical Wrist: A *spherical joint* permits arbitrary rotation for one link with respect to the other one. In our 3-DOF manipulator arm, we chose three orthogonal rotational axes, shown in Fig. 7(C), where $l_1 = l_2 = 0.1$ m, $\theta_{1,0} = 0$, and $\theta_{2,0} = \theta_{3,0} = \pi/2$. This example demonstrates that Algorithm 8 can assign origami joints to avoid collisions, even with three theoretically collocated joints.



J_i	a_i	α_i	d_i	θ_i
1	0	0	l_0	$\theta_{1,0} + q_1$
2	0	0	$l_1 + q_2$	0
3	0	0	l_2	0

J_i	a_i	α_i	d_i	θ_i
1	l_0	0	0	$\theta_{1,0} + q_1$
2	l_1	0	0	$\theta_{2,0} + q_2$
3	l_2	0	0	$\theta_{3,0} + q_3$
4	l_3	0	0	0

J_i	a_i	α_i	d_i	θ_i
1	0	0	l_1	$\theta_{1,0} + q_1$
2	0	$\pi/2$	0	$\theta_{2,0} + q_2$
3	0	$\pi/2$	0	$\theta_{3,0} + q_3$
4	0	$\pi/2$	l_2	0

Fig. 7. Kinegami results of a (A) Cylindrical, (B) Planar, and (C) Spherical manipulator. Each of the figures have subplots: (a) Coordinate frames and the D-H representation. The green cylinder represents the revolute joint, and the hexahedron represents the prismatic joint. (b) Crease pattern generated by Kinegami. The green shadowed parts indicate the joints, and the remainders indicate the links. The gray hatched area indicates the adhesive area to create the tubular sheet. The crease pattern for the tuck section of the elbow fitting is not shown for simplicity. (c) Folded state of the origami robot.

B. Self-Intersection and Compactness

As discussed in Section V-B, Algorithm 8 does not guarantee a nonself-intersecting linkage. We illustrate this issue and a potential solution by designing a 6-DOF PUMA arm (D-H specification from [47]), providing empirical support that Algorithm 9 will generate a physically valid kinegami chain over Algorithm 8. In Fig. 8(a), the centerline of the robot associated with Algorithm 8 incurs a self-intersection between two of its links, so the robot is not feasible. In contrast, in Fig. 8(b), the centerline created with Algorithm 9 has no intersection since it is rerouted via additional waypoints and is consequently longer. Of course, neither of the outputs of these greedy algorithms is the optimal joint placement, as we now demonstrate by contrast with a human-generated design that also illustrates the user-friendly aspects of the earlier pipeline steps.

Instead of locating the joints through line 2 in Algorithm 10, the user can directly specify joint centroid frames $\mathcal{O}_c$ as an

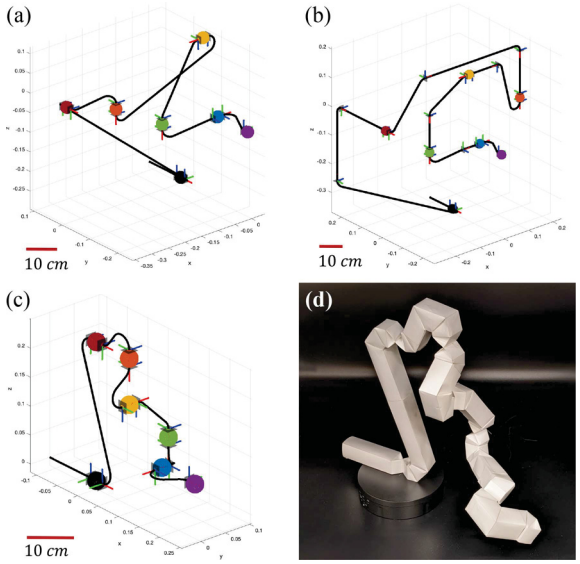


Fig. 8. Kinegami PUMA arm: Schematic drawing of the Kinegami output with joint placement using (a) Algorithm 8, (b) Algorithm 9, and (c) Human-specified joint location. Joint spheres are shown colored, with purple at the end effector. Frames without spheres indicate the waypoints in (b). The centerline of the links is in black. (d) Folded Kinegami from (c).

input. Using this "human-specified joint location" method, it is possible to place the joints closer to each other while maintaining a non-self-intersecting centerline, as shown in Fig. 8(c) and (d). Additional higher DOF joints as discussed in Section III-C could also be used to reduce the volume of the kinematic chain by combining multiple joints with coincident joint frames together. Given its known combinatorial complexity, heuristic approaches to the problem of optimizing feasible joint placement subject to the CSC path validity represent a very inviting open research domain.

C. Demonstration of Actuation

Finally, we show how the compliant joints can be used to store and release energy in a 1-DOF catapult. The robot weighs 0.3 kg, with the paper kinematic chain structure (with $\{n_s, r\} = \{4, 0.05\ \text{m}\}$) contributing 50% of the total mass. The joint centroid frames of the revolute joint and the end effector are assigned manually. We also provide intermediate waypoints $\mathcal{W}_1, \ldots, \mathcal{W}_4$ (see Section V-B) that the origami structure must pass through to form a broad base, as shown in Fig. 9(a).

The revolute joint ($q_m = \pi/2,\ n_z = 2$) is actuated by a tendon-driven servo motor with a latch mechanism adapted from [43]. One end of the tendon is attached to the distal side of the revolute joint, and the other is attached to the latch mechanism located at the base, as shown in Fig. 9(b). When the servo motor shaft turns, the tendon between the distal and the proximal end of the revolute joint is shortened and thus bends the revolute joint towards the tendon side. Once the servo motor rotates to the desired angle, the tendon is let loose, thus releasing the energy stored in the compliant revolute joint and launching the payload. The time-lapse photos of the catapult throwing a 3-g ping-pong ball and a 50-g rubber ball are shown in Fig. 9(c) and

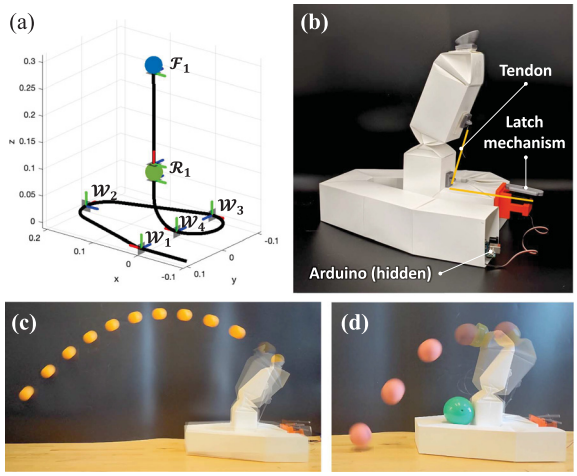


Fig. 9. Origami catapult. (a) Schematic drawing of the joint assignment. (b) Folded catapult with actuator attached. The time-lapse photos of the catapult motion throwing a (c) 3-g ping-pong ball with an interval of 1/24 s and a (d) 50-g rubber ball with an interval of 1/12 s.

(d). Using "Tracker" (https://physlets.org/tracker/) to measure the projectile trajectories, we estimate that 35 mJ of energy is transferred to the rubber ball by the compliant revolute joint in 66.7 ms, resulting in an average mechanical power output of 525 mW (i.e., a robot power density of roughly 1.75 W/kg).

VIII. Conclusion

This article established the design feasibility and demonstrated the rapid fabrication of an origami serial robot mechanism generated from well-established kinematic specifications. Work now underway aims to prove the remaining conjecture required to guarantee that the algorithm we presented will always generate a viable (nonself-intersecting) kinematic chain whose reachable poses form an open set within the joint configuration space. Similarly, inspired by the present empirical demonstration of tunably compliant joints, ongoing work aims to develop a careful stiffness model that can be used to prove our conjecture that extensions of these algorithms can yield folded structures with arbitrarily specified compliance.

Origami-inspired compliant structures open up opportunities for increasing the agility of a robot by more effectively distributing actuator power spatially in the robot and temporally throughout a given task [36]. This article offered a suggestive illustration of that idea by using a low mass, low power, conventional actuator to load energy into a 1-DOF Kinegami catapult, achieving a 525-mW transfer of 35 mJ into a ballistic payload that was 16% of the robot's mass. The integration of novel smart material actuators [57], [58] with the flat sheet substrates underlying the compliant robot structure [7] as a potential method for increasing the robot power density will be the focus of future work.

Appendix A
Second Moment of Inertia of a Prism Tube

Given a n_s-sided regular polygon of circumradius r and vertices parameterized as $(v_{i,x}, v_{i,y}) = (r\cos(\frac{2\pi i}{n_s}), r\sin(\frac{2\pi i}{n_s}))$,

its second moment of inertia with respect to the x-axis [59] is

$$I_x(n_s, r) = \frac{n_s r^4}{48}\left(4\sin\left(\frac{2\pi}{n_s}\right) + \sin\left(\frac{4\pi}{n_s}\right)\right). \quad \text{(A1)}$$

The perimeter of the regular polygon is $2n_s r\cos(\frac{\pi(n_s-2)}{2n_s})$. Assuming the perimeter is a fixed constant L, its circumscribed circle radius can be calculated to be $r_L = \frac{L}{2n_s}\csc\frac{\pi}{n_s}$.

Now assume a thin wall regular polygon prism with n_s sides, a circumscribed circle of the neutral line of the wall of radius r_L, and a wall thickness of t. The second moment of inertia of this thin-wall regular polygon prism is

$$I_{x,t}(n_s, r_L, t) := I_x\left(n_s, r_L + \tfrac{t}{2}\right) - I_x\left(n_s, r_L + \tfrac{t}{2}\right) \quad \text{(A2)}$$

which is a monotonically increasing function. Thus, $I_{x,t}$ gets larger as n_s grows.

ACKNOWLEDGMENT

The authors would like to thank C.-P. Huang, J. D. Caporale, M. Kvalheim, S. Roberts, T. Zhang, T. Topping, V. Vasilopoulos, and Y. Wu for their input on the article. The authors would also like to thank D. Krieger for her administrative support.

REFERENCES

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of origami robots," *Nature Rev. Mater.*, vol. 3, no. 6, pp. 101–112, 2018.
[2] P. S. Sreetharan, J. P. Whitney, M. D. Strauss, and R. J. Wood, "Monolithic fabrication of millimeter-scale machines," *J. Micromech. Microeng.*, vol. 22, no. 5, 2012, Art. no. 055027.
[3] M. T. Tolley, S. M. Felton, S. Miyashita, D. Aukes, D. Rus, and R. J. Wood, "Self-folding origami: Shape memory composites activated by uniform heating," *Smart Mater. Struct.*, vol. 23, no. 9, 2014, Art. no. 094006 .
[4] S. Miyashita, S. Guitron, M. Ludersdorfer, C. R. Sung, and D. Rus, "An untethered miniature origami robot that self-folds, walks, swims, and degrades," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1490–1496.
[5] A. Firouzeh, Y. Sun, H. Lee, and J. Paik, "Sensor and actuator integrated low-profile robotic origami," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4937–4944.
[6] J. T. Karras et al., "Pop-up mars rover with textile-enhanced rigid-flex PCB body," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5459–5466.
[7] Z. Zhakypov and J. Paik, "Design methodology for constructing multimaterial origami robots and machines," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 151–165, Feb. 2018.
[8] L. A. Bowen, C. L. Grames, S. P. Magleby, L. L. Howell, and R. J. Lang, "A classification of action origami as systems of spherical mechanisms," *J. Mech. Des.*, vol. 135, no. 11, 2013, Art. no. 111008.
[9] S. G. Faal, F. Chen, W. Tao, M. Agheli, S. Tasdighikalat, and C. D. Onal, "Hierarchical kinematic design of foldable hexapedal locomotion platforms," *J. Mechanisms Robot.*, vol. 8, no. 1, 2016, Art. no. 011005 .
[10] J. Angeles and F. C. Park, "Design and performance evaluation," in *Springer Handbook of Robotics*, 2nd ed. Berlin, Germany: Springer, 2016, ch. 16.
[11] D. Rus and C. Sung, "Spotlight on origami robots," *Sci. Robot.*, vol. 3, no. 15, 2018, Art. no. eaat0938.
[12] R. J. Lang, "A computational algorithm for origami design," in *Proc. ACM Annu. Symp. Comput. Geometry*, 1996, pp. 98–105.
[13] E. D. Demaine and T. Tachi, "Origamizer: A practical algorithm for folding any polyhedron," in *Proc. 33rd Int. Symp. Comput. Geometry*, vol. 77, 2017, pp. 34:1–34:16.
[14] J. Mitani, "A design method for 3D origami based on rotational sweep," *Comput.-Aided Des. Appl.*, vol. 6, no. 1, pp. 69–79, 2009.
[15] A. M. Mehta and D. Rus, "An end-to-end system for designing mechanical structures for print-and-fold robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1460–1465.
[16] A. Schulz et al., "Interactive Robogami: An end-to-end system for design of robots with ground locomotion," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1131–1147, 2017.
[17] S. Wu, Q. Ze, J. Dai, N. Udipi, G. H. Paulino, and R. Zhao, "Stretchable origami robotic arm with omnidirectional bending and twisting," *Proc. Nat. Acad. Sci.*, vol. 118, no. 36, 2021, Art. no. e2110023118.
[18] A. Firouzeh and J. Paik, "An under-actuated origami gripper with adjustable stiffness joints for multiple grasp modes," *Smart Mater. Struct.*, vol. 26, no. 5, 2017, Art. no. 055035 .
[19] J. Santoso, E. H. Skorina, M. Luo, R. Yan, and C. D. Onal, "Design and analysis of an origami continuum manipulation module with torsional strength," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2098–2104.
[20] H. Yasuda, T. Tachi, M. Lee, and J. Yang, "Origami-based tunable truss structures for non-volatile mechanical memory operation," *Nature Commun.*, vol. 8, no. 1, pp. 1–7, 2017.
[21] W.-H. Chen et al., "A programmably compliant origami mechanism for dynamically dexterous robots," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2131–2137, Apr. 2020.
[22] E. T. Filipov, T. Tachi, and G. H. Paulino, "Origami tubes assembled into stiff, yet reconfigurable structures and metamaterials," *Proc. Nat. Acad. Sci.*, vol. 112, no. 40, pp. 12321–12326, 2015.
[23] J. T. Overvelde, J. C. Weaver, C. Hoberman, and K. Bertoldi, "Rational design of reconfigurable prismatic architected materials," *Nature*, vol. 541, no. 7637, pp. 347–352, 2017.
[24] C. Sung and D. Rus, "Foldable joints for foldable robots," *J. Mechanisms Robot.*, vol. 7, no. 2, 2015, Art. no. 021012 .
[25] C. Sung, E. D. Demaine, M. L. Demaine, and D. Rus, "Edge-compositions of 3D surfaces," *J. Mech. Des.*, vol. 135, no. 11, 2013, Art. no. 111001.
[26] L. L. Howell, "Compliant mechanisms," in *21st Century Kinematics*. Berlin, Germany: Springer, 2013, pp. 189–216.
[27] C. Qiu, V. Aminzadeh, and J. S. Dai, "Kinematic analysis and stiffness validation of origami cartons," *J. Mech. Des.*, vol. 135, no. 11, 2013, Art. no. 111004.
[28] C. Sung and D. Rus, "Automated fabrication of foldable robots using thick materials," *Springer Proceedings in Advanced Robotics*. Berlin, Germany: Springer, 2018, pp. 253–266.
[29] K. Lynch and M. Mason, "Dynamic underactuated nonprehensile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1996, vol. 2, pp. 889–896.
[30] T. T. Topping, G. Kenneally, and D. E. Koditschek, "Quasi-static and dynamic mismatch for door opening and stair climbing with a legged robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1080–1087.
[31] S. Seok et al., "Design principles for energy-efficient legged locomotion and implementation on the MIT cheetah robot," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1117–1129, Jun. 2015.
[32] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 6295–6301.
[33] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robot. Automat. Lett.*, vol. 1, no. 2, pp. 900–907, Jul. 2016.
[34] B. Hannaford, P.-H. Marbot, P. Buttolo, M. Moreyra, and S. Venema, "Scaling of direct drive robot arms," *Int. J. Robot. Res.*, vol. 15, no. 5, pp. 459–472, 1996.
[35] J. Zhang et al., "Robotic artificial muscles: Current progress and future perspectives," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 761–781, Jun. 2019.
[36] M. Ilton et al., "The principles of cascading power limits in small, fast biological and engineered systems," *Science*, vol. 360, no. 6387, 2018, Art. no. eaao1082.
[37] J. M. Duperret, G. D. Kenneally, J. Pusey, and D. E. Koditschek, "Towards a comparative measure of legged agility," in *Exp. Robot.* Cham, Switzerland: Springer, 2016, vol 109, pp. 3–16.
[38] R. Wood, R. Nagpal, and G.-Y. Wei, "Flight of the roboBees," *Sci. Amer.*, vol. 308, no. 3, pp. 60–65, 2013.
[39] D. W. Haldane, K. C. Peterson, F. L. G. Bermudez, and R. S. Fearing, "Animal-inspired design and aerodynamic stabilization of a hexapedal millirobot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3279–3286.
[40] Z. Zhakypov, M. Falahi, M. Shah, and J. Paik, "The design and control of the multi-modal locomotion origami robot, tribot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4349–4355.
[41] H. Yuan, J. Pikul, and C. Sung, "Programmable 3-D surfaces using origami tessellations," in *Proc. 7th Int. Meeting Origami Sci., Math., Educ.*, 2018, pp. 893–906.
[42] W.-H. Chen, S. Misra, J. D. Caporale, D. E. Koditschek, S. Yang, and C. R. Sung, "A tendon-driven origami hopper triggered by proprioceptive contact detection," in *Proc. IEEE Int. Conf. Soft Robot.*, 2020, pp. 373–380.

[43] J. Carlson, J. Friedman, C. Kim, and C. Sung, "Rebound: Untethered origami jumping robot with controllable jump height," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 10089–10095.
[44] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. London, U.K.: Butterworth-Heinemann, 2004.
[45] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. London, U.K.: Pearson, 2005.
[46] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mechanics*, vol. 22, no. 2, pp. 215–221, 1955.
[47] K. J. Waldron and J. Schmiedeler, "Kinematics," in *Springer Handbook of Robotics*, 2nd ed. Berlin, Germany: Springer, 2016, ch. 2, pp. 11–33.
[48] R. J. Lang, *Twists, Tilings, and Tessellations: Mathematical Methods for Geometric Origami*. Boca Raton, FL, USA: AK Peters/CRC Press, 2017.
[49] A. D. Luca and W. J. Book, "Robots with flexible elements," *Springer Handbook of Robotics*, 2nd ed. Berlin, Germany: Springer, Chapter 11, 2016, pp. 243–279.
[50] T. C. Hull, *Origametry: Mathematical Methods in Paper Folding*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
[51] S. Hota and D. Ghose, "Optimal geometrical path in 3D with curvature constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 113–118.
[52] R. L. Gieseking, "Gieseking Origami/ crease patterns: Crimp-bends," 2016. [Online]. Available: http://rebecca.gieseking.us/2016/12/crease-patterns-crimp-bends/
[53] T. Tachi, "3D origami design based on tucking molecule," *Origami*, vol. 4, pp. 259–272, 2009.
[54] R. J. Lang, *Origami Design Secrets: Mathematical Methods for an Ancient Art*. Boca Raton, FL, USA: AK Peters/CRC Press, 2011.
[55] T. A. Evans, R. J. Lang, S. P. Magleby, and L. L. Howell, "Rigidly foldable origami twists," *Origami*, vol. 6, no. 1, pp. 119–130, 2015.
[56] V. Gabrel, A. R. Mahjoub, R. Taktak, and E. Uchoa, "The multiple Steiner TSP with order constraints: Complexity and optimization algorithms," *Soft Comput.*, vol. 24, pp. 17957–17968, 2020.
[57] C. Kim, A. Chien, M. Tippur, and C. Sung, "Fabrication and characterization of I-cord knitted SMA actuators," in *Proc. IEEE 4th Int. Conf. Soft Robot.*, 2021, pp. 379–386.
[58] Y. Gao, J. Liu, and S. Yang, "Scalable fabrication of liquid crystal elastomeric filaments as high-performance actuators," in *Proc. ACM Symp. Comput. Fabr.*, 2019.
[59] D. Hally, "Calculation of the moments of polygons," Defence Research Establishment Suffield Ralston (Alberta), Tech. Rep. DREA-TM-87/209, 1987.

Wei-Hsi Chen (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from National Taiwan University, Taipei, Taiwan, in 2011 and 2013, respectively, and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2022.

He is currently a Postdoctoral Researcher with the Department of Electrical and Systems Engineering. His research interests include understanding the embodied intelligence of robotics through bio-inspired and computational design and fabrication.

Woohyeok Yang received the B.S. degree in physics from Kookmin University, Seoul, South Korea, in 2018, and the M.S. degree in mechanical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2021.

He is currently a Process Engineer with LG Energy Solution Michigan, Inc. Hazel Park, MI, USA. His research interests include robotic applications and battery manufacturing.

Lucien Peach received the B.S. degree in mechanical engineering, in 2022 from the University of Pennsylvania, Philadelphia, PA, USA, where he is currently working toward the M.S. degree in robotics as part of the Accelerated Master's Program.

His research interests include soft robotics, path planning, and robotic control.

Daniel E. Koditschek (Life Fellow, IEEE) received the B.S. degree in engineering and applied science and the M.S. and Ph.D. degrees in electrical engineering, in 1981 and 1983, respectively, all from Yale University, New Haven, CT, USA.

He joined the faculty at Yale in 1983 and moved to the University of Michigan, Ann Arbor, in 1992. In 2005, he was appointed Chair of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia, PA, USA, in which he currently serves as the Alfred Fitler Moore Professor, with secondary appointments in the Departments of Computer and Information Science as well as Mechanical Engineering and Applied Mechanics. His research interests include robotics and the application of dynamical systems theory to intelligent machines.

Dr. Koditschek was the recipient of the 2016 IEEE Robotics and Automation Society Pioneer Award and the 2017 Penn Heilmeier Award for Excellence in Research.

Cynthia R. Sung (Member, IEEE) received the B.S. degree in mechanical engineering from Rice University, Houston, TX, USA, in 2011, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2016.

In 2017, she joined the University of Pennsylvania as the Gabel Family Term Assistant Professor with the Department of Mechanical Engineering and Applied Mechanics, with secondary appointments with the Departments of Computer and Information Science and of Electrical and Systems Engineering. Her research interests include computational design and fabrication for robotic systems, with a particular focus on origami-inspired and compliant robots.

Dr. Sung was the recipient of the 2019 NSF CAREER award and the 2020 Johnson & Johnson WiSTEM2D award.