Advanced Robotics

ISSN: 0169-1864 (Print) 1568-5535 (Online) Journal homepage: www.tandfonline.com/journals/tadr20

Kinematic modeling and formability analysis of revolved bodies formed by origami waterbomb units based on a chain-like layer-building method

Guanyu Chen, Songhao Liu & Xuelin Wu

To cite this article: Guanyu Chen, Songhao Liu & Xuelin Wu (2024) Kinematic modeling and formability analysis of revolved bodies formed by origami waterbomb units based on a chain-like layer-building method, Advanced Robotics, 38:11, 730-744, DOI: 10.1080/01691864.2024.2353141

To link to this article: https://doi.org/10.1080/01691864.2024.2353141

Published online: 27 May 2024.

ADVANCED ROBOTICS
2024, VOL. 38, NO. 11, 730–744
https://doi.org/10.1080/01691864.2024.2353141

FULL PAPER

OPEN ACCESS Check for updates

Kinematic modeling and formability analysis of revolved bodies formed by origami waterbomb units based on a chain-like layer-building method

Guanyu Chen[a], Songhao Liu[b] and Xuelin Wu[b]

[a]Samueli School of Engineering, University of California, Los Angeles, CA, USA; [b]School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, USA

ABSTRACT

Origami patterns play a critical role in the field of soft and reconfigurable robotics. The revolved body formed by waterbomb units is also widely used in robot design. The kinematic model plays a key role in understanding motion characteristics and is vital for dynamic modeling and control of robots based on origami. However, while the deformation of origami patterns is well-studied on the finite element level, the assembly of rigid origami patterns is rarely explored on the kinematic level. Therefore, we propose a chain-like assembly method for constructing revolved bodies using rigid waterbomb units, ensuring crease status unalternation, collision avoidance, axis existence, and seamless assembly. We explore the formability of waterbomb units and investigate the resulting revolved body with some layer and column numbers of waterbomb units. Results demonstrate that increasing the number of columns does not necessarily provide more space for building layers. Additionally, they reveal boundaries of the layer and column numbers for constructing the revolved body and highlight the impact of the aspect ratio and configuration of the waterbomb units on the formability of the revolved body. This method can provide insights for origami-based robot research and can be extended to model other origami patterns.

ARTICLE HISTORY
Received 11 July 2023
Revised 27 December 2023
Accepted 18 April 2024

KEYWORDS
Kinematic model; origami; waterbomb unit assembly; chain-like layer-building method; formability

1. Introduction

Origami, the Japanese art of paper folding, has been explored and applied to various fields of engineering, such as novel structure design, and robotics [1–3]. The outstanding features of origami such as softness and reconfigurability can be achieved by robotic origamis (Robogamis) through multiple degrees of freedom (DoFs) [4]. The three-dimensional (3D) morphologies of origami structures can be transformed from predetermined patterns of creases on a two-dimensional sheet to achieve a specific behavior or task with embedded sensors and actuators [5–7]. Origami robots can be easy to store, transport, deploy into functional robots when required, and mold to and interact with an object's surface due to its low thickness, lightweight, and high flexibility [8]. Besides, the fabrication process of folding from two-dimensional patterns to three-dimensional structures can be easy and fast [9–12]. Those advantages make origami robotics competitive to traditional robotics especially in situations such as rehabilitation [13], and rescue [14,15] where safety should be ensured and the external environment is dynamic.

There are various types of origami patterns such as Miura-Ori pattern [16], augmented square twist (AST) [17], kresling origami [18], waterbomb [19], and so on, which can form diverse 3D structures for different working purposes. The waterbomb is a traditional origami pattern, and it has two types of bases: an eight-crease base and a six-crease base [20]. The tessellation comprised of six-crease bases can form a tube-like revolved body called a magic ball whose volume can change with its length [20]. Some robots are taking inspiration from this, such as the origami swimmer via jet propulsion [21], the deformable wheel robot [22], the worm robot [23], and so on.

Kinematic models for origami robots can be of great significance in predicting their motion based on their geometry design. An origami modular robot called Mori has been developed in [8], and the kinematic model of the robot that forms a waterbomb unit has been studied to implement a ball-balancing experiment to showcase its manipulation ability. However, since only one waterbomb unit was considered in [8] without coupling it with other units, the kinematic model was easy to build. Besides, the deformation of origami patterns

CONTACT Guanyu Chen gychen98@ucla.edu

is commonly explored through finite element methods, which can entail substantial computational costs when compared to kinematic-level methods. Furthermore, the kinematic model can play an essential role in the dynamic analysis and control of origami robots. According to [24], a decentralized control architecture has been applied to a multi-material and layer-fabricated origami modular robot based on the kinematic analysis, making the robot able to prescribe specific motions. Therefore, a method is proposed in this paper to build the kinematic model of origami robots when they transform into more complicated configurations whose units are highly coupled with each other.

In this paper, we take the origami magic ball as the research sample and propose a chain-like method to build the kinematic model of the entire revolved body layer by layer after building the unit kinematic model of the waterbomb pattern. Furthermore, we study the ability to form both the waterbomb units and the entire revolved body, while ensuring that the status of creases on the pattern do not swap, the waterbomb units do not self-collide and mutually collide, the axis of revolution exists, and the units on the same layer can be assembled seamlessly. We find that for the formability of each vertex unit, the aspect ratio of the unit can influence the acceptable range of the dihedral angles to form the units successfully, and one dihedral angle can also affect the feasible range of the other dihedral angles in a unit. The formability analysis for the revolved body shows that one revolved body can be successfully built with bounded combinations of layer number and column number of the waterbomb units. Besides, an increasing number of columns does not necessarily indicate that more layers can be built. The formability of the revolved body is also affected by both the aspect ratio of waterbomb units and their configurations on the initial layer. Although the paper focuses on the waterbomb pattern, the methodology possesses adaptability for diverse origami patterns and origami-inspired robotics.

The structure of the paper is organized as follows. Section 2 describes the kinematic modeling process, which incorporates the kinematic model of each type of waterbomb unit, determination of the axis of revolution, unit assembly, and chain-like layer-building method. Section 3 explores the formability of each type of waterbomb unit and the entire revolved body. Section 4 validates the method proposed and the formability through simulation, and discusses the results and limitations. Section 5 concludes the achievements accomplished in this paper and addresses future work.

2. Method

The waterbomb pattern studied in this paper is based on three types of six-crease vertex units (waterbomb units), shown in Figure 1(a), whose six creases $c_1, c_2, \ldots, c_6$ radiate from the central vertex shown as red, blue, and green dots for type-1, type-2, and type-3 vertexes, respectively. The type-3 vertex is the inverted version of the type-2 vertex and is essentially identical to the type-2 vertex. Each type of vertex unit and its structural parameters are shown in Figure 1(b–g). The dash-dotted line represents that the crease is in mountain status, while the dotted line represents that the crease is in valley status. β_1, β_2, $\ldots$, β_6 are the dihedral angles corresponding to creases $c_1, c_2, \ldots c_6$. The creases in mountain or valley status are assigned dihedral angles within the open range from 0 to 180 degrees. If the dihedral angles are beyond the range, the state of the creases swaps. We can say not only that the entire pattern consists of all three types of vertex units with overlapped panels but also that it consists of solely the rows of the type-1 vertex units without overlaps but with misalignment on adjacent rows. Therefore, the total size of the pattern can be determined by the length of the horizontal and vertical edges of the type-1 vertex unit. The magic ball is a tube-like revolved body fabricated by folding the pattern and connecting the left and right sides of the pattern.

A layer of waterbomb units is not equivalent to a row of units. When we say that a new layer is added, it means that a new vertex unit is added to the previous layer in each column of vertexes. In contrast, when we say that a new row is added, it means that a new row of the rectangular type-1 vertex units is added to the previous row of type-1 vertex units. This paper considers layer-building, as different types of vertex units share common dihedral angles, forming the basis for our implemented method.

One column of the waterbomb units can be considered to build the kinematic model, and can then be revolved to complete the kinematic model of the entire body for convenience. The column that begins with the type-1 vertex unit is used in this paper. Note that the columns of the units are not equivalent to the columns of vertexes. One column of units consists of three columns of the vertexes, and one column of vertexes is shared by the adjacent two columns of units, as shown in Figure 1(a).

2.1. Assumption

In this paper, it is assumed that the material can only be rigidly folded rather than bent or stretched. Besides,

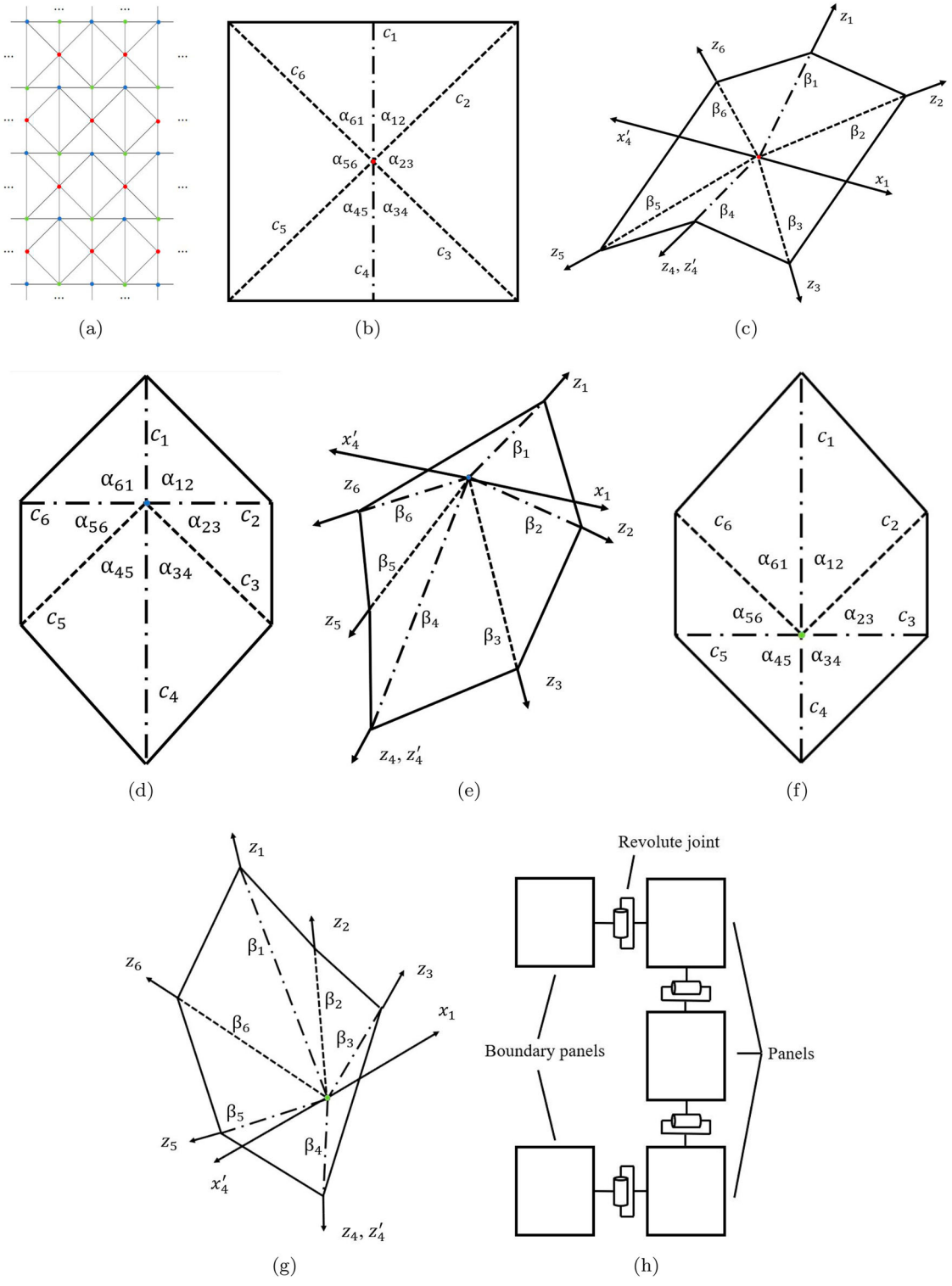


Figure 1. Waterbomb pattern with six-crease vertexes and the coordinate frames attached to each type of vertex. (a) Waterbomb pattern. If the vertex units whose central vertexes are represented by the blue dots and green dots on the top form the first layer, then the vertex units whose central vertexes are represented by the adjacent green dots and red dots on each column of vertexes form the second layer. (b) Type-1 vertex unit. (c) Coordinate frames attached to the type-1 vertex. (d) Type-2 vertex unit. (e) Coordinate frames attached to the type-2 vertex. (f) Type-3 vertex unit. (g) Coordinate frames attached to the type-3 vertex. (h) Simplified schematics of one side of the unit.

the thickness of the sheet is assumed to be zero for convenience [20]. Readers might also refer to [20] to learn how to build a kinematic model for the waterbomb pattern when considering the thickness and refer to [25] to find out the material and dimension of the waterbomb unit that can provide a satisfying stiffness for the entire body.

2.2. kinematic model of each type of vertex

The kinematic model for each type of vertex is built using the Denavit–Hartenberg convention (DH convention). The coordinate frames for all types of vertex units are built, as shown in Figure 1(c,e,g). Note that the x axis of frame 1 is perpendicular to the plane that bisects the dihedral angle β_1, and the x axis of frame $4'$ is perpendicular to the plane that bisects the dihedral angle β_4. The x axis of the rest of the frames x_{i+1} is determined by $z_i \times z_{i+1}$. The related DH parameters are shown in Table 1 where α is the angle of rotation from z_i axis to z_{i+1} axis about x_{i+1} axis, and θ is the angle of rotation from x_i axis to x_{i+1} axis about z_i axis. Since all coordinate frames are built at the central vertex in each vertex unit, the two DH parameters related to the displacement and offset should be zero and not shown in Table 1. The orientation of frame $i+1$ based on frame i can be expressed in Equation (1).

$$\mathbf{R}^{\mathbf{i}}_{\mathbf{i+1}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{1}$$

$$\boldsymbol{L} = \begin{cases} \begin{bmatrix} \frac{E_2}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} & \frac{E_2}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} \end{bmatrix} & \text{for type-1 vertex unit} \\ \begin{bmatrix} \frac{E_2}{2} & \frac{E_1}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} & E_2 & \frac{\sqrt{E_1^2+E_2^2}}{2} & \frac{E_1}{2} \end{bmatrix} & \text{for type-2 vertex unit} \\ \begin{bmatrix} E_2 & \frac{\sqrt{E_1^2+E_2^2}}{2} & \frac{E_1}{2} & \frac{E_2}{2} & \frac{E_1}{2} & \frac{\sqrt{E_1^2+E_2^2}}{2} \end{bmatrix} & \text{for type-3 vertex unit} \end{cases} \tag{4}$$

where E_1 and E_2 are the horizontal edge length and vertical edge length of the type-1 vertex unit, respectively.

Frame 1 is regarded as the base frame. The orientation of frame i based on frame 1, $\boldsymbol{R}^{\mathbf{1}}_{\boldsymbol{i}}$ can be acquired in Equation (2).

$$\boldsymbol{R}^{\mathbf{1}}_{\boldsymbol{i}} = \boldsymbol{R}^{\mathbf{1}}_{\mathbf{2}}\boldsymbol{R}^{\mathbf{2}}_{\mathbf{3}} \cdots \boldsymbol{R}^{\boldsymbol{i-1}}_{\boldsymbol{i}} \tag{2}$$

where $\boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{i}} \in SO(3)$ is the orientation of frame i based on frame j. The position of the node at the end of the ith crease based on frame 1, $\boldsymbol{P}^{\mathbf{1}}_{\boldsymbol{i}} \in \mathbb{R}^3$ can be obtained using Equation (3). The orientation of the creases aligns with the z axis in the corresponding frames.

$$\boldsymbol{P}^{\mathbf{1}}_{\boldsymbol{i}} = l_i \boldsymbol{R}^{\mathbf{1}}_{\boldsymbol{i}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3}$$

where $l_i \in \mathbb{R}$ is the length of the ith crease in each type of the vertex units, and it can be easily obtained using Equation (4). Note that l_i is also the ith element in $\boldsymbol{L}$ for each type of the vertex units.

2.3. Constraints of the kinematic model

Since the revolved body is focused on in this paper, the left and right parts of the vertex units should be symmetric to each other. Therefore, the counterpart angles on the

Table 1. DH parameters for the three types of vertex units.

	Type-1 vertex unit		Type-2 vertex unit		Type-3 vertex unit	
From frame	θ	α	θ	α	θ	α
1 to 2	$\pi - \frac{\beta_1}{2}$	α_{12}	$\pi - \frac{\beta_1}{2}$	α_{12}	$\pi - \frac{\beta_1}{2}$	α_{12}
2 to 3	$-\pi + \beta_2$	α_{23}	$\pi - \beta_2$	α_{23}	$-\pi + \beta_2$	α_{23}
3 to 4	$-\pi + \beta_3$	α_{34}	$-\pi + \beta_3$	α_{34}	$\pi - \beta_3$	α_{34}
4 to $4'$	$\pi - \frac{\beta_4}{2}$	0	$\pi - \frac{\beta_4}{2}$	0	$\pi - \frac{\beta_4}{2}$	0

two sides of the plane of symmetry in each vertex unit should equal each other, as shown in Equation (5).

$$\begin{aligned}
&\text{For type-1 vertex unit,}\\
&\alpha_{12} = \alpha_{61} = \alpha_{34} = \alpha_{45} = atan2(E_1, E_2),\\
&\alpha_{23} = \alpha_{56} = 2atan2(E_2, E_1)\\
&\text{For type-2 vertex unit,}\\
&\alpha_{12} = \alpha_{61} = \frac{\pi}{2}, \quad \alpha_{23} = \alpha_{56} = atan2(E_2, E_1),\\
&\alpha_{34} = \alpha_{45} = atan2(E_1, E_2)\\
&\text{For type-3 vertex unit,}\\
&\alpha_{12} = \alpha_{61} = atan2(E_1, E_2),\\
&\alpha_{23} = \alpha_{56} = atan2(E_2, E_1), \quad \alpha_{34} = \alpha_{45} = \frac{\pi}{2}
\end{aligned} \tag{5}$$

where α_{ij} is the angle between the ith crease and the jth crease. These angles are constant and can be predetermined based on the geometry of the origami pattern.

Besides, the x axis of frame $4'$ based on frame 1 should not have any component in y and z direction in frame 1, as shown in Equation (6), if only half of the vertex unit is considered.

$$R^1_{4'x|y} = 0 \quad \text{and} \quad R^1_{4'x|z} = 0 \tag{6}$$

where $R^1_{4'x|y}$ and $R^1_{4'x|z} \in \mathbb{R}$ mean the y and z component of the x axis of frame $4'$ based on frame 1, respectively.

The dihedral angles, which can cause different configurations of each vertex unit, are variables. There are six dihedral angles in each vertex unit. Since the vertex units are symmetric, only the four dihedral angles on one side of and directly on the plane of symmetry are considered to build the kinematic model of the units for convenience. The simplified mechanism schematics for half of the units are shown in Figure 1(h) where the boundary panels are the panels on the other side (left side) that connect the current side (right side) of the units, and the revolute joints represent the corresponding creases c_1, c_2, c_3, and c_4 on the right side of the units. Therefore, the total degrees of freedom in the half unit, *DoFs* can be calculated using Gruebler's Equation below.

$$DoF = 3(n-1) - 2l - h = 4 \tag{7}$$

where $n = 5$ is the number of panels playing the role of linkages, $l = 4$ is the number of lower pairs, and $h = 0$ is the number of higher pairs. Therefore, one-half of each vertex unit only has 4 degrees of freedom. Based on the two constraints shown in Equation (6), two of the four dihedral angles in the half unit can be used to solve for the other two, as shown in Equation (8), and further to solve for the complete configuration of the vertex unit. In other words, there are only 2 degrees of freedom due to the two constraints.

$$\beta_i, \beta_j, \boldsymbol{Ori}, \boldsymbol{Pos}, flag = solve(\beta_m, \beta_n, \boldsymbol{E}, type) \tag{8}$$

where β_m and β_n are two dihedral angles used to solve for the other two angles β_i and β_j in a half vertex unit. If the dihedral angles are beyond the range from 0 to 180 degrees, it means that the corresponding crease status swaps. This is not studied in this paper, since the vertex units in this case are not in a normal configuration that can form a magic ball. ***Ori*** is a 3×6 matrix where each column represents the orientation of a certain crease c_i based on frame 1, $\boldsymbol{z_i} \in \mathbb{R}^3 (i = 1, 2, 3, \ldots 6)$ in each vertex unit. ***Pos*** is a 3×7 matrix whose columns represent the positions of the six vertexes at the end of each crease, $\boldsymbol{P^1_i} \in \mathbb{R}^3 (i = 1, 2, 3, \ldots 6)$ and the position of the central vertex based on frame 1, $\boldsymbol{P^1_7} \in \mathbb{R}^3$ in one vertex unit, *flag* can be true or false, which indicates whether the equation system of the kinematic model formed by Equation (6) can be solved, $\boldsymbol{E} = \begin{bmatrix} E_1 & E_2 \end{bmatrix}$ reflects the size of the type-1 vertex unit, and $type \in \{1, 2, 3\}$ indicates the type of the vertex unit to be solved.

From the geometric intuition of the vertex units, there could be two solution sets for most cases, one when all creases remain at their original status and the other one when the status of some creases swaps. It can be proved easily by imagining a plane formed by crease c_1 and c_4, Pn_1 (also known as the plane of symmetry) under all the assumptions and constraints mentioned above. For instance, assuming that β_1 and β_2 are already known, crease c_3 is now fixed. Hence, the plane formed by crease c_3 and c_4, Pn_2 can be considered as a bar whose one end is pinned on c_3 and the other end lies on Pn_1. There are only two ways to achieve that for most cases, being that Pn_2 contacts Pn_1 with an upward tilt or downward tilt. In this paper, we assume that creases maintain their status without change. Consequently, there exists only one feasible set of solutions within the open range of 0 to 180 degrees for the dihedral angles, and singular configurations are not achievable.

In this paper, the equation system based on Equation (6) is solved using the toolbox *lsqnonlin* in MATLAB R2024A since it is efficient for solving nonlinear equations with boundaries. The function searches the solution in a range from 0 to 180 degrees.

2.4. Determination of the axis of revolution

Since each row of the pattern consists of many type-1 vertex units without overlaps, the node at the end of c_2 and c_6 in all type-1 vertex units in a row can form a polygon, as shown in Figure 2(a). The axis of revolution

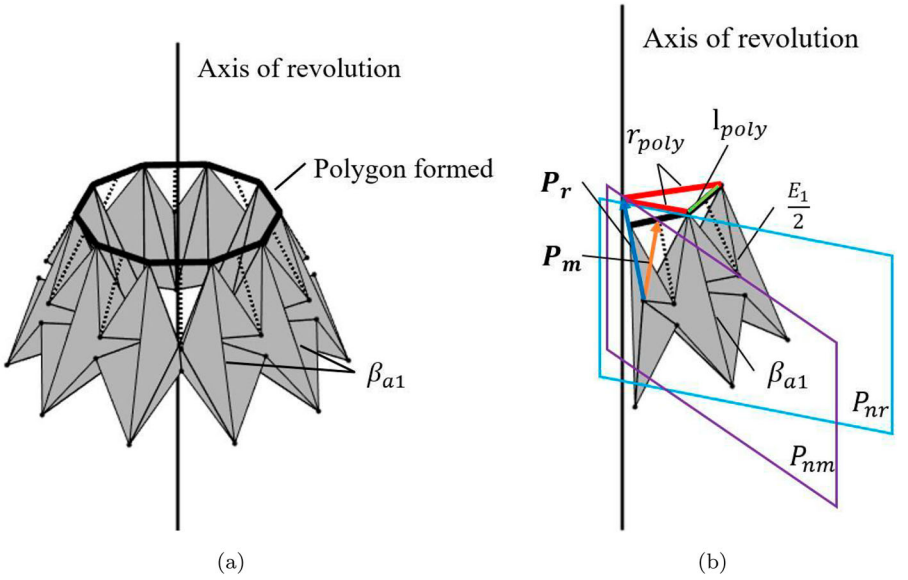


Figure 2. Schematics to find out the axis of revolution. (a) The first row of the type-1 vertex units. (b) Two adjacent type-1 vertex units with key parameters.

will go through the center of the polygon and be perpendicular to the plane where the polygon lies in. The parameters that are known in advance are the number of columns C, the number of layers R, horizontal and vertical length of the type-1 vertex unit E_1 and E_2 respectively, and two independent dihedral angles in the half type-1 vertex unit in the first row (in this paper β_1 and β_2 are set to be known initially). These parameters will determine the dimension and geometry of the magic ball. The two known dihedral angles are used to discover the configuration of that unit, so all four angles in the half unit can be obtained initially. To discover the axis of revolution and dihedral angle β_{a1}, two type-1 vertex units in the first row can be used, as shown in Figure 2(b). Note that β_{a1} is the dihedral angle of the crease shared by the two adjacent type-1 vertex units in the first row, and knowing β_{a1} is vital for the layer-building process of the entire body kinematics.

Firstly, the half length of the diagonal of the polygon, r_{poly} can be acquired in Equation (9).

$$r_{poly} = \frac{l_{poly}}{2\sin\left(\frac{\pi}{C}\right)} = \frac{E_1 \sin\left(\frac{\beta_1}{2}\right)}{\sin\left(\frac{\pi}{C}\right)} \tag{9}$$

where l_{poly} is the edge length of the polygon. The position of the midpoint on the edge of the polygon $\boldsymbol{P_m}$ can be obtained in Equation (10).

$$\boldsymbol{P_m} = \boldsymbol{P_2} + \frac{1}{2}(\boldsymbol{P_6} - \boldsymbol{P_2}) \tag{10}$$

where $\boldsymbol{P_i}$ is the position of the node at the end of c_i $(i = 1, 2, \ldots, 6)$ in the left unit in Figure 2(b).

Assume that one point at the intersection of the axis of revolution and the polygonal plane is $\boldsymbol{P_r} = (x_r, y_r, z_r)$ in 3-dimensional space. The normal vectors of the two useful planes Pn_m and Pn_r are figured out in Equations (11) and (12).

$$\hat{\boldsymbol{n}}_{\boldsymbol{m}} = \frac{\boldsymbol{P_m} \times \boldsymbol{z_1}}{\|\boldsymbol{P_m} \times \boldsymbol{z_1}\|} \tag{11}$$

where $\hat{\boldsymbol{n}}_{\boldsymbol{m}}$ is the unit normal vector of the plane Pn_m formed by $\boldsymbol{P_m}$ and c_1, and $\boldsymbol{z_i}$ is the orientation of c_i $(i = 1, 2, \ldots, 6)$ in the left unit in Figure 2(b).

$$\hat{\boldsymbol{n}}_{\boldsymbol{r}} = \frac{(\boldsymbol{P_2} - \boldsymbol{P_3}) \times (\boldsymbol{P_2} - \boldsymbol{P_r})}{\|(\boldsymbol{P_2} - \boldsymbol{P_3}) \times (\boldsymbol{P_2} - \boldsymbol{P_r})\|} \tag{12}$$

where $\hat{\boldsymbol{n}}_{\boldsymbol{r}}$ is the unit normal vector of the plane Pn_r formed by the vector $\boldsymbol{P_2} - \boldsymbol{P_3}$ and the vector $\boldsymbol{P_2} - \boldsymbol{P_r}$.

The orientation of the axis of revolution $\hat{\boldsymbol{n}}_{\boldsymbol{rev}}$ is the intersection line between the plane Pn_m and Pn_r, as shown in Equation (13).

$$\hat{\boldsymbol{n}}_{\boldsymbol{rev}} = \frac{\hat{\boldsymbol{n}}_{\boldsymbol{m}} \times \hat{\boldsymbol{n}}_{\boldsymbol{r}}}{\|\hat{\boldsymbol{n}}_{\boldsymbol{m}} \times \hat{\boldsymbol{n}}_{\boldsymbol{r}}\|} \tag{13}$$

The unknown position $\boldsymbol{P_r}$ can be figured out using the following geometric relationships in Equation (14) where $\hat{\boldsymbol{n}}_{\boldsymbol{rev}}$ can be substituted by Equation (13). After that, $\hat{\boldsymbol{n}}_{\boldsymbol{rev}}$ can be obtained.

$$\begin{aligned} \left\| \frac{(\boldsymbol{P_m} - \boldsymbol{P_r}) \times \hat{n}_{rev}}{\|\hat{\boldsymbol{n}}_{\boldsymbol{rev}}\|} \right\| &= r_{poly} \cos\left(\frac{\pi}{C}\right) \\ \left\| \frac{(\boldsymbol{P_2} - \boldsymbol{P_r}) \times \hat{n}_{rev}}{\|\hat{\boldsymbol{n}}_{\boldsymbol{rev}}\|} \right\| &= r_{poly} \\ \hat{\boldsymbol{n}}_{\boldsymbol{rev}} \cdot (\boldsymbol{P_2} - \boldsymbol{P_r}) &= 0 \end{aligned} \tag{14}$$

After determining the axis of revolution, the dihedral angle β_{a1} shown in Figure 2 can be figured out using the following equations. This paves the way for the chain-like layer-building process.

$$\hat{\boldsymbol{n}}_{\boldsymbol{p}} = \frac{(\boldsymbol{P_7} - \boldsymbol{P_2}) \times (\boldsymbol{P_7} - \boldsymbol{P_3})}{\|(\boldsymbol{P_7} - \boldsymbol{P_2}) \times (\boldsymbol{P_7} - \boldsymbol{P_3})\|} \tag{15}$$

where $\hat{\boldsymbol{n}}_{\boldsymbol{p}}$ is the unit normal vector of the plane formed by crease c_2 and c_3, and remember that $\boldsymbol{P_7}$ is the position of the central vertex in the left unit in Figure 2(b). Therefore,

$$\beta_{a1} = 2\cos^{-1}\left(\frac{\hat{\boldsymbol{n}}_{\boldsymbol{p}} \cdot \hat{\boldsymbol{n}}_{\boldsymbol{r}}}{\|\hat{\boldsymbol{n}}_{\boldsymbol{p}}\|\|\hat{\boldsymbol{n}}_{\boldsymbol{r}}\|}\right) \tag{16}$$

Those mentioned above is packaged in a function called $\boldsymbol{Ori_{rev}}, \boldsymbol{P_r}, \beta_{a1}, flag_{rev} = findrev(\boldsymbol{Pos}, \boldsymbol{Ori}, \beta_1, C, \boldsymbol{E})$. where $flag_{rev}$ can be *True* or *False* to determine whether the equation system for the axis of revolution can be solved, *Pos* and *Ori* are the positions of all nodes and orientation of all creases in the first type-1 vertex unit, respectively, and β_1 is the dihedral angle of crease c_1 in the first type-1 vertex unit in a column of vertex units.

2.5. Vertex unit assembly

The adjacent vertex units share some common creases. Remember that all positions of nodes and orientations of creases in a vertex unit are based on frame 1, and all the units that have not been assembled yet share the same frame 1 as the first vertex unit. All assembled units are based on frame 1 of the first vertex unit (frame f1) in a column of vertex units in this paper. Thus, for the vertex units that have not been assembled yet,

$$\boldsymbol{Pos^{f1}} = \boldsymbol{Pos^{1}}, \boldsymbol{Ori^{f1}} = \boldsymbol{Ori^{1}} \tag{17}$$

where $\boldsymbol{Pos^k}$ and $\boldsymbol{Ori^k}$ are the positions of all nodes and orientations of all creases in one vertex unit based on frame k. Therefore, to assemble the vertex units in a column with symmetry-related constraints, only two conditions should be satisfied. First, the z_1 axis which is the orientation of the crease c_1 of the current vertex unit should be re-orientated and aligned with the $-z_4$ axis which is the opposite orientation of the crease c_4 of the previously assembled vertex unit based on frame f1. Note that the reason for the negative sign in front of z_4 is that the shared creases have opposite orientations when adjacent units are assembled. The relevant rotation matrix found in frame f1, $\boldsymbol{R_{as}^{f1}}$ can be easily obtained using the two orientations mentioned above. Besides, the two creases should coincide with each other, so the position of the vertex at the end of the crease c_4 of the previously assembled unit, $\boldsymbol{P_{prev4}^{f1}}$ should be the same as the position of the central vertex of the current unit. Therefore, the new orientation of all creases $\boldsymbol{Ori_{as}^{f1}}$ and position of all nodes $\boldsymbol{Pos_{as}^{f1}}$ in the current unit after assembly can be expressed in the following based on frame f1.

$$\boldsymbol{Ori_{as}^{f1}} = \boldsymbol{R_{as}^{f1}}\boldsymbol{Ori^{f1}} \tag{18}$$

$$\boldsymbol{Pos_{as}^{f1}} = \boldsymbol{R_{as}}\boldsymbol{Pos^{f1}} + \boldsymbol{P_{prev4}^{f1}} \tag{19}$$

where $\boldsymbol{Ori^{f1}}$ and $\boldsymbol{Pos^{f1}}$ are the orientations of all creases and positions of all nodes before assembly, respectively, and $\boldsymbol{P_{prev4}^{f1}}$ is the position of node at the end of crease c_4 of the previously assembled unit based on frame f1. This assembly process can be packaged into the function $\boldsymbol{Ori_{as}^{f1}}, \boldsymbol{Pos_{as}^{f1}} = assemble(\boldsymbol{z_{prev4}}, \boldsymbol{P_{prev4}^{f1}}, \boldsymbol{Ori^{f1}}, \boldsymbol{Pos^{f1}})$ where $\boldsymbol{z_{prev4}}$ is $\boldsymbol{z_4}$ of the previously assembled unit.

2.6. Chain-like layer-building method

Since two of the four dihedral angles in half vertex unit are employed to calculate the remaining two, with these two angles obtainable from the preceding vertex unit in the current column and the prior unit in the adjacent column, this process can be carried out using one column of vertex units starting with the type-1 unit and its adjacent column. The mechanism works in the following way, as shown in Figure 3. The angle $\beta_{i,(n,k)}$ is known as the ith dihedral angle of the nth vertex unit in the kth column of vertexes. The cluster of blocks on the left is the column of vertexes that starts with the type-1 vertex unit, while the cluster of blocks on the right is the adjacent column of vertexes. Each row in a cluster represents a half-vertex unit and consists of four dihedral angles. The shaded blocks of the dihedral angle mean the angle can be known initially or from the previous vertex units. Since the adjacent vertex units share some common creases, the solid arrow means the inter-column transfer of angles, and the dotted arrow means the intra-column transfer of angles between different vertex units. It can be seen from Figure 3 that there can be two known angles in each row of the two clusters, which indicates that it is feasible to build the kinematic model of the entire body given two proper dihedral angles for the first vertex unit. The chain-like layer-building process is shown in Algorithm 1 which shows how a new layer is added using the function *addlayer*(...). The input $\boldsymbol{Pos_{n,k}}$ and $\boldsymbol{Ori_{n,k}}$ are the positions of all vertexes and the orientations of all creases in the nth vertex unit of the kth column of vertexes. $\boldsymbol{E}$ contains the length of the horizontal and vertical edges of the type-1 vertex unit, and $n+1$ is the index of the layer to be built next. The dihedral angles in the input are defined in Figure 3. The function outputs the positions of all vertexes and the orientations of all creases in the

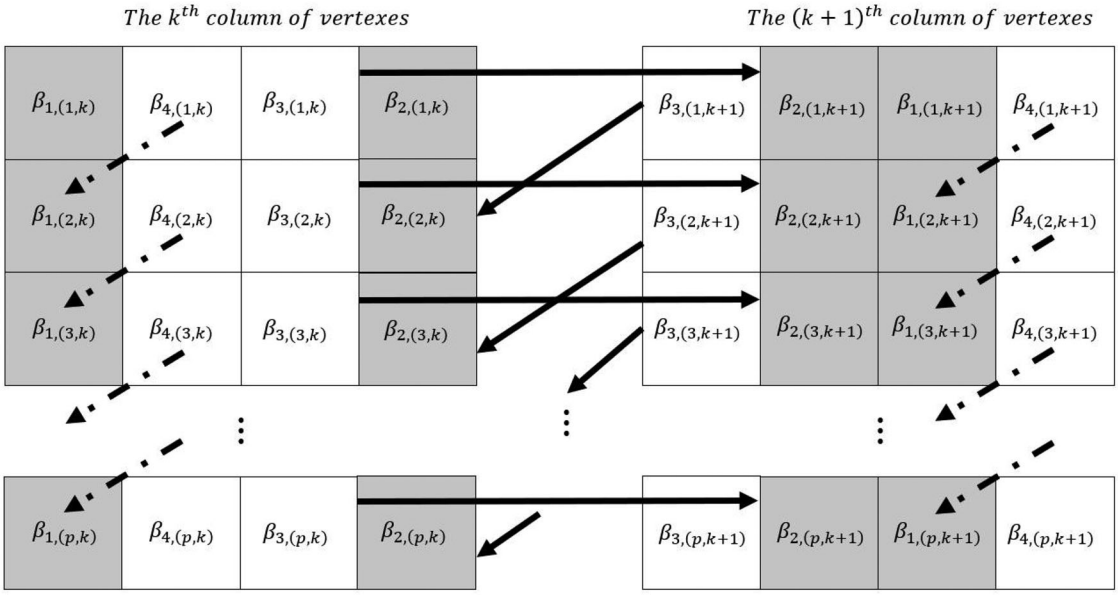


Figure 3. Working mechanism of the chain-like layer-building process.

next vertex unit of the kth column of vertexes, $\boldsymbol{Pos_{n+1,k}}$ and $\boldsymbol{Ori_{n+1,k}}$, some dihedral angles that are useful for the further building process, and a flag $flag_{lr}$ which indicates whether the new layer is built successfully with the consideration of collision. The process to build the entire revolved body is shown in Algorithm 2 and packaged in the function *buildbody*. The inputs include the number of layers to be built R, the number of columns of the revolved body C, the type-1 vertex unit dimension $\boldsymbol{E}$, and two initial dihedral angles $\beta_{3,(n,k+1)}$ and $\beta_{4,(n,k+1)}$. The functions *checkunit* and *checkbody* are used to check the formability of vertex units and the entire body which is discussed in Section 3.

3. Formability of vertex units and revolved body

3.1. Formability of vertex units

From Section 2.6, we know that the dihedral angles β_1 and β_2 in every vertex unit are either provided initially or can be obtained from the previous vertex unit to calculate β_3 and β_4 of the current unit. The conditions that determine whether a unit can be formed should depend not only on whether the kinematic model of the unit is solvable but also on whether self-collision could occur. Based on this, the function *checkunit* to check the formability of the three types of vertex units is shown in Algorithm 3.

In Algorithm 3, *flag* obtained from the function *solve* indicates whether the kinematic model is solvable for the dihedral angles within the range from 0 to 180 degrees, $B_{1\times4} = [\beta_1 \ \ \beta_2 \ \ \beta_3 \ \ \beta_4]$ is a 1×4 matrix which contains the four dihedral angles in the half vertex unit, $Ori_{3\times6}$ is a 3×6 matrix which contains the column vectors representing the orientation of all creases in a unit, and $type \in \{1, 2, 3\}$ indicates which type the vertex unit belongs to. The output of the function is a bool indicator.

The algorithm first checks whether the kinematic model of vertex units is solvable using the input *flag*, and then checks if self-collision occurs. There can be two situations of collision for the vertex units. Situation 1 is that the panels on the left side collide with those on the right side, as shown in Figure 4(a). Therefore, the creases c_2 and c_3 in all types of vertex units should stay on their original side. It is interesting to find that the kinematic model of vertex units is often not solvable when this situation of collision happens, probably due to the limits of dihedral angles. However, to ensure safety, we still add this rule. Situation 2 is that the panels on one side collide mutually, as shown in Figure 4(b). For type-1 vertex unit, this situation can occur, since creases c_2 and c_3 are both in valley status and crease c_4 could collide with crease c_1 from geometric intuition. Therefore, the y-component of crease c_4 should be smaller than zero to ensure that the panels on one side will not collide mutually. For type-2 and type-3 vertex units, situation 2 could not occur, since creases c_2 and c_3 are in opposite status.

3.2. Formability of the revolved body

The formability of the revolved body can depend on individual unit formability, inter-unit collision avoidance, and seamless unit assembly. Therefore, one of the preconditions for the success of the revolved body formation is to ensure that all units can be successfully formed. To

Algorithm 1 Chain-like layer-building process to build layer $n+1$ from layer n

Input: $\boldsymbol{Pos_{n,k}}, \boldsymbol{Ori_{n,k}}, \boldsymbol{E}, n+1, \beta_{4,(n,k)}, \beta_{3,(n,k+1)}, \beta_{4,(n,k+1)}$

Output: $\boldsymbol{Pos_{n+1,k}}, \boldsymbol{Ori_{n+1,k}}, \beta_{3,(n+1,k)}, \beta_{4,(n+1,k)}, \beta_{3,(n+1,k+1)}, \beta_{4,(n+1,k+1)}, flag_{lr} = addlayer(...)$

$\beta_{1,(n+1,k)} = \beta_{4,(n,k)}$ ▷ Dihedral angles for shared creases

$\beta_{2,(n+1,k)} = \beta_{3,(n,k+1)}$

$type_1 = mod(n+1, 3)$

$type_2 = mod(n+3, 3)$

if $type_1 = 0$ **then**

  $type_1 = 3$

end if

if $type_2 = 0$ **then**

  $type_2 = 3$

end if

$\beta_{3,(n+1,k)}, \beta_{4,(n+1,k)}, \boldsymbol{Ori_{n+1,k}}, \boldsymbol{Pos_{n+1,k}}, flag = solve(\beta_{1,(n+1,k)}, \beta_{2,(n+1,k)}, \boldsymbol{E}, type_1)$

▷ Solve for the vertex unit on the $(n+1)$th row and kth column of vertexes

$Indicator_1 =$ ▷ Check formability of the unit

$checkunit(flag, [\beta_{1,(n+1,k)}, \beta_{2,(n+1,k)}, \beta_{3,(n+1,k)}, \beta_{4,(n+1,k)}], \boldsymbol{Ori_{n+1,k}}, type_1)$

$\boldsymbol{Ori_{n+1,k}}, \boldsymbol{Pos_{n+1,k}} = assemble(\boldsymbol{Ori_{n,k}}(:,4), \boldsymbol{Pos_{n,k}}(:,4), \boldsymbol{Ori_{n+1,k}}, \boldsymbol{Pos_{n+1,k}})$

▷ Attach the new unit to the previous unit

$\beta_{1,(n+1,k+1)} = \beta_{4,(n,k+1)}$ ▷ Dihedral angles for shared creases

$\beta_{2,(n+1,k+1)} = \beta_{3,(n+1,k)}$

$\beta_{3,(n+1,k+1)}, \beta_{4,(n+1,k+1)}, \boldsymbol{Ori_{n+1,k+1}}, \boldsymbol{Pos_{n+1,k+1}}, flag =$
$solve(\beta_{1,(n+1,k+1)}, \beta_{2,(n+1,k+1)}, \boldsymbol{E}, type_2)$

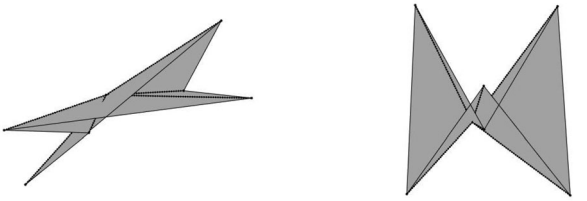
▷ Solve for the vertex unit on the $(n+1)$th row and $(k+1)$th column of vertexes

$\boldsymbol{B} = [\beta_{1,(n+1,k+1)}, \beta_{2,(n+1,k+1)}, \beta_{3,(n+1,k+1)}, \beta_{4,(n+1,k+1)}]$

$Indicator_2 = checkunit(flag, \boldsymbol{B}, \boldsymbol{Ori_{n+1,k+1}}, type_2)$

▷ Check formability of the unit

$flag_{lr} = Indicator_1 \cap Indicator_2$



(a) (b)

Figure 4. Collision situations. (a) Situation 1 collision. (b) Situation 2 collision.

avoid inter-unit collision among the units at counterpart positions, the nodes that are at the concave positions or the bottom side of one vertex unit on the pattern should be figured out, since they will be more likely to collide with each other from geometric intuition. The central node is at the concave position and the node at the end of the crease c_3 is at the bottom for the type-1 units. Since the node at the end of the crease c_4 is at the bottom but at a convex position, its inter-unit collision is not considered for the type-1 units. The node at the end of the creases c_1 and c_3 is at the concave position and the node at the end of the crease c_4 is at the bottom for the type-2 units. The node at the end of the crease c_2 is at the concave position and the node at the end of the crease c_4 is at the bottom for the type-3 units. Since the revolved body is similar to an ellipsoidal shape, the nodes in the last and the first vertex units in a column will probably collide first with their counterparts. Additionally, since the units that can form a loop on the previous layer do not guarantee that the units on the next layer can form a loop, whether the units can be attached seamlessly on each layer should be checked.

The function *checkbody* in Algorithm 4 where the nodes on only one side of the symmetric plane in vertex units are considered, is used to check the formability of the revolved body. The inputs are the number of layers and columns, R and C respectively, and the positions of all nodes in a column of vertex units, $[\boldsymbol{Pos_{tot}}]_{3\times 7R}$. The output is a bool indicator. Based on the geometric intuition of the revolved body, if the nodes collide with their counterparts, they will overpass the axis of revolution. Remember the axis of revolution has been moved to be aligned with the z axis in the world frame in Section 2.6. The position of a node that is always on

Algorithm 2 Chain-like layer-building process for the entire body

Input: R, C, $\boldsymbol{E}$, $\beta_{1,(1,k)}$, $\beta_{2,(1,k)}$
Output: $\boldsymbol{Pos_{tot}}, \boldsymbol{Ori_{tot}} = buildbody(R, C, \boldsymbol{E}, \beta_{1,(1,k)}, \beta_{2,(1,k)})$

```
β3,(1,k), β4,(1,k), Ori1,k, Pos1,k, flag = solve(β1,(1,k), β2,(1,k), E, 1)
flag = checkunit(flag, [β1,(1,k), β2,(1,k), β3,(1,k), β4,(1,k)], Ori1,k, 1)
if flag = False then
    Terminate the whole process.
else                                   ▷ Find the axis of revolution using the first unit
    Orirev, Pr, βa1, flagrev = findrev(Pos1,k, Ori1,k, β1,(1,k), C, E)
    if flagrev = False then
        Terminate the whole process.
    else
        β2,(1,k+1) = β3,(1,k)
        β1,(1,k+1) = βa1
        if R > 1 then
            β3,(1,k+1), β4,(1,k+1), Ori1,k+1, Pos1,k+1, flag =
            solve(β1,(1,k+1), β2,(1,k+1), E, 3)
                                   ▷ All parameters required for the layer-building process are obtained
            flag =
            checkunit(flag, [β1,(1,k+1), β2,(1,k+1), β3,(1,k+1), β4,(1,k+1)], Ori1,k+1, 3)
            if flag = False then
                Terminate the whole process.
            else                           ▷ Start the layer-building process
                Postot = Pos1,k
                Oritot = Ori1,k
                for i = 2 : 1 : R do
                    Posi,k, Ori i,k, β3,(i,k), β4,(i,k), β3,(i,k+1), β4,(i,k+1), flaglr =
                    addlayer(Posi−1,k, Ori i−1,k, E, i, β4,(i−1,k), β3,(i−1,k+1), β4,(i−1,k+1))
                    if flaglr = False then
                        Terminate the whole process.
                    else
                        Postot = [Postot Posi,k]
                        Oritot = [Oritot Ori i,k]
                    end if
                end for
                Move the axis of revolution to align with the vector [0 0 1]^T (considered
                 as the world z axis).   ▷ Essential for the bodycheck function below
                if checkbody(R, Postot) = False then
                                   ▷ Check whether inter-unit collision occurs
                    Terminate the whole process.
                else
                    Revolve Postot and Oritot around z axis with a step of 2π/C to add
                     the positions of all the vertexes and orientations of all creases on
                     the entire body to Postot and Oritot.
                end if
            end if
        end if
    end if
end if
```

Algorithm 3 Formability of the three types of the vertex units

Input: *flag*, $B_{1\times4}$, $Ori_{3\times6}$, and *type*
Output: *Indicator* = *checkunit*(. . .)
  if flag=True **then** ▷ Check the solvability of the kinematic model of vertex units
    if $\boldsymbol{Ori}(1,2) > 0$ and $\boldsymbol{Ori}(1,3) > 0$ and ($type \neq 1$ or $\boldsymbol{Ori}(2,4) < 0$) **then** ▷ Check self-collision
      Indicator = *True*
    else
      Indicator = *False*
      ▷ The kinematic model can be solved, but the panels will self-collide
    end if
  else
    Indicator = *False* ▷ The kinematic model cannot be solved
  end if

Algorithm 4 Formability of the revolved body

Input: R, C, $[\boldsymbol{Pos_{tot}}]_{3\times7R}$
Output: *Indicator* = *checkbody*(. . .)
  Indicator = *True* ▷ Check the first vertex unit in a column of units
  if $\langle \boldsymbol{Pos_{tot}}(1:2,2), \boldsymbol{Pos_{tot}}(1:2,i)\rangle < 0$ $(i = 3, 7)$ **then**
    Indicator = *False*
  end if
  if $R > 1$ **then** ▷ Check the last vertex unit in a column of units
    if $mod(R,3) = 1$ **then** ▷ If type-1 vertex unit is the last one
      if $\langle \boldsymbol{Pos_{tot}}(1:2,2), \boldsymbol{Pos_{tot}}(1:2,i)\rangle < 0$ $(i = 7R-4, 7R)$ **then**
        Indicator = *False*
      end if
    else if $mod(R,3) = 2$ **then** ▷ If type-2 vertex unit is the last one
      if $\langle \boldsymbol{Pos_{tot}}(1:2,2), \boldsymbol{Pos_{tot}}(1:2,i)\rangle < 0$ $(i = 7R-6, 7R-4, 7R-3)$ **then**
        Indicator = *False*
      end if
    else ▷ If type-3 vertex unit is the last one
      if $\langle \boldsymbol{Pos_{tot}}(1:2,2), \boldsymbol{Pos_{tot}}(1:2,i)\rangle < 0$ $(i = 7R-5, 7R-3)$ **then**
        Indicator = *False*
      end if
    end if
  end if
  for $j = 1:1:R$ **do** ▷ Check if the units can be assembled seamlessly
    if $\|\boldsymbol{R_{zu}}(\frac{2\pi}{C})\boldsymbol{Pos_{tot}}(:,7j-2) - \boldsymbol{Pos_{tot}}(:,7j-4)\| > \delta$ *or* $\|\boldsymbol{R_{zu}}(\frac{2\pi}{C})\boldsymbol{Pos_{tot}}(:,7j-1) - \boldsymbol{Pos_{tot}}(:,7j-5)\| > \delta$ **then**
      Indicator = *False*
      Break
    end if
  end for

one side of the revolved body and projected onto the $x-y$ plane, $\boldsymbol{Pos_{tot}}(1:2,2)$ can be easily determined. Let $\boldsymbol{Pos_{tot}}(1:2,i)$ be the position of nodes that are at the concave position or the bottom of the vertex unit, and projected onto the $x-y$ plane(i is used to locate a node of a unit on a certain layer). Hence, if the inner product $\langle \boldsymbol{Pos_{tot}}(1:2,2), \boldsymbol{Pos_{tot}}(1:2,i)\rangle$ is less than zero, some parts in vertex units will overpass the axis of revolution and collide with the counterparts. To check if the assembly of the units on the same layer can be implemented seamlessly, the positions of the nodes at the end of creases c_5 and c_6, $\boldsymbol{Pos_{tot}}(:,7j-2)$ and $\boldsymbol{Pos_{tot}}(:,7j-1)$ respectively, are rotated by $2\pi/C$ about the global z axis using $\boldsymbol{R_{zu}}(\frac{2\pi}{C})$ and compared with the positions of the nodes at the end of creases c_3 and c_2, $\boldsymbol{Pos_{tot}}(:,7j-4)$ and $\boldsymbol{Pos_{tot}}(:,7j-5)$ respectively(j means the jth layer). If the difference between their positions is smaller than a certain threshold δ, the units can be assembled seamlessly.

4. Simulation and discussion

The algorithms above are validated in MATLAB. The tolerance level of the solutions for the kinematic model we set is five orders of magnitude smaller than the size of the vertex units. In all simulations below, the length of the vertical edge of type-1 units E_2 is set as 60 mm. If the length is at different scales, then the tolerance for the solver should be changed correspondingly to keep the solutions stable. For the formability of vertex units, the range of dihedral angles with different aspect ratios of E_1/E_2 when the units can be successfully formed is displayed in Figure 5. When the dihedral angles equal 0 or 180 degrees (singular positions), the corresponding creases are in neither mountain status nor valley status. The requirement of crease status unalteration becomes invalid in these cases. Thus, the singularity situation is not considered in this paper, and the dihedral angles β_1 and β_2 have a range from 0.1 to 179.9 degrees in Figure 5. The aspect ratio of E_1/E_2 can have a great effect on the feasible range of the dihedral angles to form a unit successfully, and one dihedral angle also affects the feasible range of the other dihedral angles in the unit.

Then, the entire revolved body construction is simulated, and the formability of the body is shown in Figure 6.

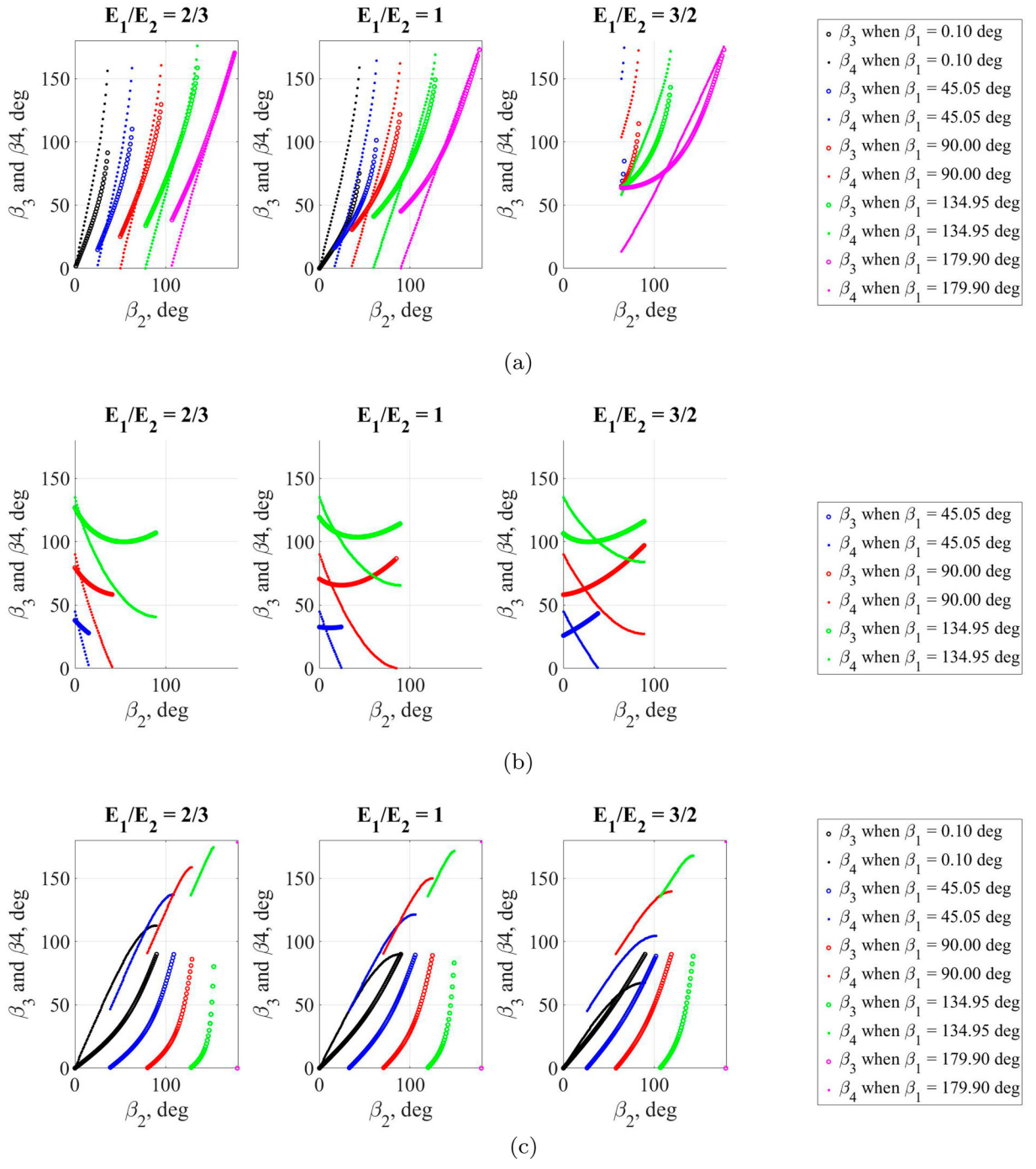


Figure 5. Formability of vertex units. For type-1 units, if the aspect ratio of E_1/E_2 is smaller or greater than 1 (the units will become slim or wide, respectively), the feasible range of β_2 generally becomes narrower. With increasing β_1, the feasible range of β_2 generally shifts to the right. When the aspect ratio of E_1/E_2 is 3/2, the units cannot be formed with β_1 of 0.1 degrees and β_2 from 0.1 to 179.9 degrees. For type-2 units, the units cannot be formed with β_1 of 0.1 or 179.9 degrees and β_2 from 0.1 to 179.9 degrees. Moreover, with increasing β_1, the feasible range of β_2 becomes wider. For type-3 units, there are few valid solutions for the kinematic model when β_1 equals 179.9 degrees. With increasing β_1, the feasible range of β_2 shifts to the right. The aspect ratio of E_1/E_2 has little impact on the feasible range of the dihedral angles. (a) For type-1 vertex units. (b) For type-2 vertex units and (c) For type-3 vertex units.

The number of layers and columns, and the configuration of the top vertex unit can influence the formability of the entire body. There is a left boundary for the column number and an upper boundary for the layer number. The reason could be that the counterpart units are likely to collide with an increasing layer number, as shown in Figure 7(a–c), or with a decreasing column number, as shown in Figure 7(d–f). Besides, with the increasing

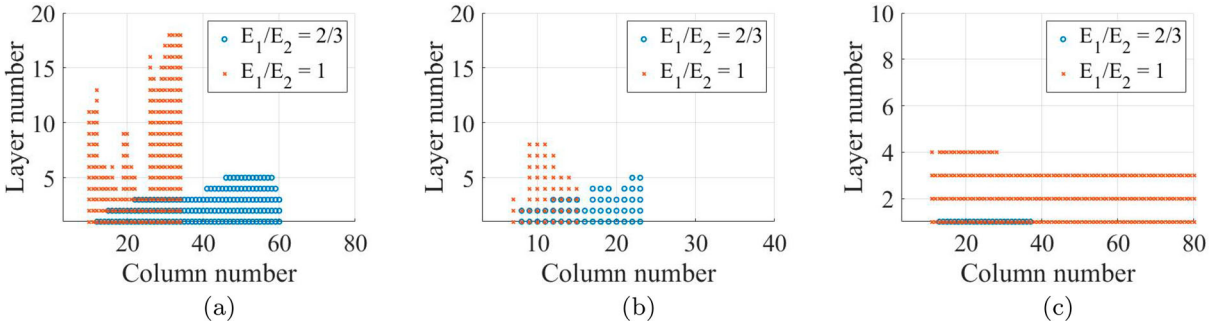


Figure 6. Formability of the revolved body. The threshold δ shown in Algorithm 4 is set to be 1×10^{-4} mm in this paper. β_1 and β_2 are the two dihedral angles of the type-1 vertex units on the first layer since they are regarded as the inputs in this paper, as was mentioned in the previous sections. When the aspect ratio of E_1/E_2 is 2/3, the revolved body can be formed by fewer layers but more columns for the configuration of β_1 and β_2 shown in Figure 6(a,b). In these two subfigures, there is a general increasing trend of layer number with increasing column number with a zigzag profile of the upper boundary of the layer number, since more layers can induce more probability that the units on a layer cannot be assembled seamlessly. When β_1 is 35 degrees and β_2 equals 25 degrees, the smaller aspect ratio of the units can lead to fewer combinations of layer and column numbers. When the aspect ratio of E_1/E_2 is 3/2, the revived body cannot be formed with any layer and column numbers for the values of β_1 and β_2 shown in the subfigures. (a) When $\beta_1 = 25^\circ$ and $\beta_2 = 25^\circ$. (b) When $\beta_1 = 25^\circ$ and $\beta_2 = 35^\circ$ and (c) When $\beta_1 = 35^\circ$ and $\beta_2 = 25^\circ$.

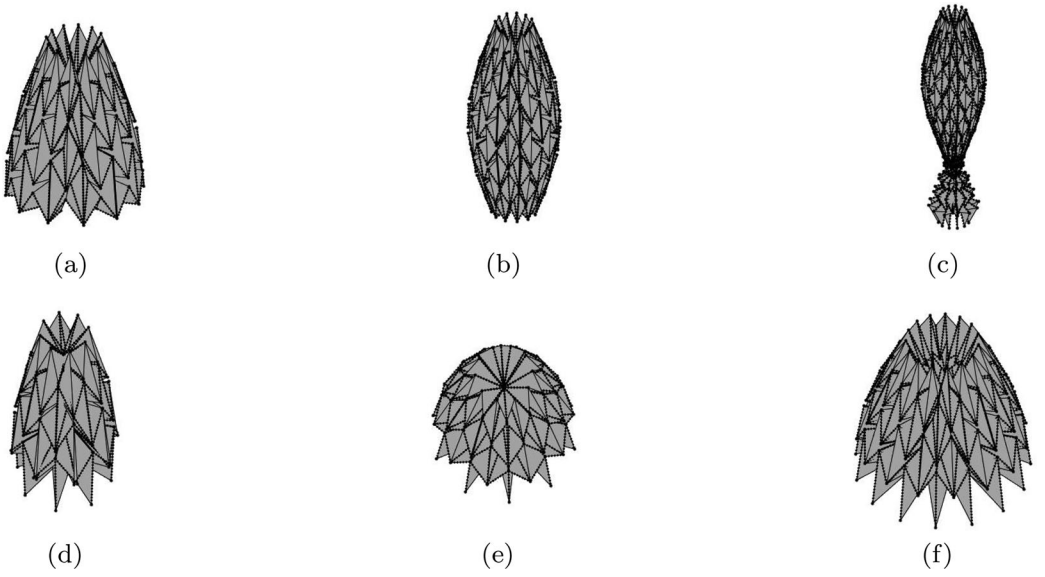


Figure 7. Entire body formation when $\beta_1 = 25^\circ$ and $\beta_2 = 25^\circ$ and $E_1/E_2 = 1$. The first row of the figure shows the body with an increasing number of layers, while the second row of the figure shows the body with an increasing number of columns. In Figure 7(c) the units at the bottom collide and extend, and this configuration is infeasible. In Figure 7(d) the top units collide mutually. The adjacent units on a layer cannot be assembled seamlessly through rotation, although this phenomenon does not occur frequently. Thus, some small misalignment of edges can be seen. This configuration is also unachievable in the real scenario. The edge misalignment and collision cases can be detected by Algorithm 4, and the building process can be terminated when the cases occur. Nevertheless, to visually illustrate these cases to the reader, we still render the complete body. The rest of the revolved bodies are collision-free. (a) 12 columns and 6 layers. (b) 12 columns and 12 layers. (c) 12 columns and 20 layers. (d) 8 columns and 5 layers. (e) 10 columns and 5 layers and (f) 15 columns and 5 layers.

number of layers, the vertex units sometimes cannot be formed successfully due to the range limits of the dihedral angles. There is also a boundary on the right side beyond which the axis of revolution could not exist. The reason can be as follows. The four nodes at the end of creases c_2, c_3, c_5, and c_6 in the type-1 units can form an isosceles trapezoid. The axis of revolution is derived using the first row of the type-1 units (a row of isosceles

trapezoids). The top-base angle of the isosceles trapezoid should range from $\pi/2 - \pi/C$ to $\pi/2 + \pi/C$ where C is the number of columns. This range narrows with larger C. If the top-base angle exceeds this range, the trapezoids cannot connect without overlapping, indicating the axis of revolution's nonexistence. The lower bound of the angle cannot be achieved due to collisions. The axis of revolution's existence is also influenced by the unit configuration on the first layer and the units' aspect ratio, as they affect the trapezoids' top-base angle. Furthermore, with the increasing column number, the polygon formed at the top of the magic ball can become larger, intuitively allowing more layers to be added without collision. However, more layers elicit more possibility that the units on a layer cannot be assembled seamlessly. Hence, it can be noticed that the upper boundary of the layer number is zigzag, as shown in Figure 6.

Assuming a zero-thickness sheet and rigid panels without bent creases simplifies the kinematic model in this paper. Using only half a unit for the model keeps the equations relatively simple, enhancing computational efficiency. However, material thickness may constrain the dihedral angle between adjacent panels of the magic ball, especially when it's small, causing interference. A higher thickness-to-length/width ratio narrows the valid range of dihedral angles. In addition, the kinematic model near the creases becomes complex for thick panels [20]. Thus, the kinematic model based on the zero-thickness assumption could lose accuracy partially in this case. Nevertheless, the influence of the thickness can be negligible when the thickness-to-length/width ratio is extremely small. The intricate kinematic model near the creases can be avoided if the kinematic analysis is applied to a rigid robot or mechanism with hinges [8]. The limitations of the method proposed in this paper include that if the magic ball is fabricated using soft materials such as plastic sheets, the assumption of rigid folding cannot be satisfied and the accuracy of the kinematic model will deteriorate. Besides, the inter-unit collision detection proposed is naive and based on geometric intuition. A more reliable collision detector can be developed in the future.

To extend the method proposed in this paper to other origami patterns, the kinematic model of each unit with symmetry-related constraints should be figured out. The tolerance level of the solutions for the kinematic model should be set according to specific requirements. Then, the degree of freedom of each unit should be calculated with symmetry-related constraints to determine how many dihedral angles are required to define the whole unit. After setting initial states, the configuration of the first unit can be obtained and the axis of revolution can be figured out based on geometric relationships. At the same time, at least one dihedral angle shared between the current and the adjacent units can be obtained to start the chain-like layer-building process. The number of shared dihedral angles for each unit should equal its degree of freedom to ensure the feasibility of this process.

5. Conclusion

In this paper, a chain-like layer-building method is proposed to build the kinematic model of the waterbomb pattern with six-crease bases. Only half a unit is considered to build the kinematic model with the assumption of zero-thickness and symmetry to improve the model's simplicity and computational efficiency. Furthermore, the formability of each type of vertex unit is figured out by discovering the range of dihedral angles with consideration of the solvability of the kinematic model and collision avoidance. The aspect ratio of a unit can affect its feasible configuration, and one dihedral angle can also affect the feasible range of the other dihedral angles in a unit. Based on this, the formability of the entire body is simulated and displayed by showing all feasible combinations of layer and column numbers. The results show that the acceptable combinations of layer and column numbers that can be used to form the revolved body successfully are bounded due to collision, unsolvability of the kinematic model, nonexistence of the axis of revolution, or disjointed edges of units on the same layer. Besides, more columns do not necessarily provide more room for building layers. The formability of the revolved body can also be affected by the aspect ratio of the vertex units and the dihedral angles of the units on the first layer. It is believed that the method proposed in this paper can contribute to the design, kinematics, and control of origami-magic-ball-based robots, and it can be extended to build kinematic models for other origami patterns. In the future, the kinematic model of asymmetric revolved bodies can be studied. The stiffness of the entire body and the transmissibility of driving force and torque from one unit to the other can also be investigated.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Guanyu Chen earned his B.E. from Southwest Jiaotong University in 2020 and his MS in Mechanical Engineering and Applied Science from The University of Pennsylvania in 2022. Currently, he is a Ph.D. candidate at the Samueli School of Engineering, University of California, Los Angeles. His research interests encompass origami robots and exoskeletons.

Songhao Liu completed his B.E. at the Beijing University of Chemical Technology in 2019 and his MS in Mechanical Engineering and Applied Science at the University of Pennsylvania in 2023. His research focuses on origami robots and robotics for rehabilitation.

Xuelin Wu earned her B.E. from the University of Mississippi in 2022. She then pursued her MS in Mechanical Engineering and Applied Science at the University of Pennsylvania, graduating in 2024. Her research interests lie in the realms of soft robots and origami robots.

References

[1] Ma J, You Z. Energy absorption of thin-walled beams with a prefolded origami pattern. Thin-Walled Struct. 2013;73:198–206. doi: 10.1016/j.tws.2013.08.001

[2] Leal ER, Dai JS. From origami to a new class of centralized 3-DOF parallel mechanisms. Paper presented at: IDETC/CIE 2007. Proceedings of the 31st Mechanisms and Robotics Conference; 2007 Sep 4–7; Las Vegas, USA.

[3] Okuzaki H, Saido T, Suzuki H, et al. A biomorphic origami actuator fabricated by folding a conducting paper. Paper presented at: World Congress on Biomimetics, Artificial Muscles and Nano-bio 2007. Journal of Physics: Conference Series; 2008 Nov 6–9; Torre Pacheco, Spain.

[4] Firouzeh A, Paik J. Robogami: a fully integrated low-profile robotic origami. ASME J Mech Robot. 2015;7:021009. doi: 10.1115/1.4029491

[5] Hawkes E, An B, Benbernou NM, et al. Programmable matter by folding. Proc Nat Acad Sci. 2010;107:12441–12445. doi: 10.1073/pnas.0914069107

[6] Benbernou N, Demaine ED, Demaine ML, et al. A universal crease pattern for folding orthogonal shapes. arXiv. 2009;0909.5388.

[7] Onal CD, Tolley MT, Wood RJ, et al. Origami-inspired printed robots. IEEE/ASME Trans Mechatron. 2015;20:2214–2221. doi: 10.1109/TMECH.2014.2369854

[8] Belke CH, Paik J. 2017 mori: a modular origami robot. IEEE/ASME Trans Mechatron. 2017;22:2153–2164. doi: 10.1109/TMECH.2017.2697310

[9] Martinez RV, Fish CR, Chen X, et al. Elastomeric origami programmable paper-elastomer composites as pneumatic actuators. Adv Funct Mater. 2012;22:1376–1384. doi: 10.1002/adfm.v22.7

[10] Onal CD, Wood RJ, Rus D. Towards printable robotics: origami-inspired planar fabrication of three-dimensional mechanisms. Paper presented at: IEEE International Conference on Robotics and Automation 2011. IEEE International Conference on Robotics and Automation; 2011 May 9–13; Shanghai, China.

[11] Felton S, Tolley M, Onal C, et al. Robot self-assembly by folding: a printed inchworm robot. Paper presented at: IEEE International Conference on Robotics and Automation 2013. IEEE International Conference on Robotics and Automation; 2013 May 6–10; Karlsruhe, Germany.

[12] Noh M, Kim SW, An S, et al. Flea-inspired catapult mechanism for miniature jumping robots. IEEE Trans Robot. 2012;28:1007–1018. doi: 10.1109/TRO.2012.2198510

[13] Song YS, Sun Y, van den Brand R, et al. Soft robot for gait rehabilitation of spinalized rodents. Paper presented at: IEEE International Conference on Robotics and Automation 2013. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2013 Nov 3–7; Tokyo, Japan.

[14] Matsuda T, Murata S. Stiffness distribution control-locomotion of closed link robot with mechanical softness. Paper presented at: ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation; 2006 May 15–19; Orlando, USA.

[15] Firouzeh A, Ozmaeian M, Alasty A, et al. An IPMC-made deformable-ring-like robot. Smart Mater Struct. 2012;21:065011. doi: 10.1088/0964-1726/21/6/065011

[16] Kaddour AS, Velez CA, Georgakopoulos SV. A deployable and reconfigurable origami reflectarray based on the Miura-Ori pattern. Paper presented at: IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting 2020. IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting; 2020 Jul 5–10; Montreal, Canada.

[17] Rubio AJ, Kaddour AS, Brown N, et al. A physically reconfigurable origami reflectarray based on the augmented square twist pattern. Paper presented at: WMCS 2021. IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS); 2021 May 18–20; Waco, USA.

[18] Liu X, Yao S, Georgakopoulos SV. Mode reconfigurable bistable spiral antenna based on kresling origami. Paper presented at: IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting 2017. IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting; 2017 Jul 9–14; San Diego, USA.

[19] Tang J, Wang X. A compliant origami mechanism driven by tendon-sheath. Paper presented at: IAEAC 2021. IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference; 2021 Mar 12–14; Chongqing, China.

[20] Chen Y, Feng H, Ma J, et al. Symmetric waterbomb origami. Proc R Soc A. 2016;472:20150846. doi: 10.1098/rspa.2015.0846

[21] Yang Z, Chen D, Levine DJ, et al. Origami-inspired robot that swims via jet propulsion. IEEE Robot Autom Lett. 2021;6:7145–7152. doi: 10.1109/LRA.2021.3097757

[22] Lee DY, Kim JS, Kim SR, et al. The deformable wheel robot using magic ball origami structure. Paper presented at: IDETC/CIE 2013. 37th Mechanisms and Robotics Conference; 2013 Aug 4–7; Portland, USA.

[23] Onal CD, Wood RJ, Rus D. An origami-inspired approach to worm robots. IEEE/ASME Trans Mechatron. 2013;18:430–438. doi: 10.1109/TMECH.2012.2210239

[24] Robertson MA, Kara OC, Paik J. Soft pneumatic actuator-driven origami-inspired modular robotic "pneumagami". Int J Rob Res. 2021;40:72–85. doi: 10.1177/0278364920909905

[25] Yellowhorse A, Howell LL. Three approaches for managing stiffness in origami-inspired mechanisms. Proc. the ASME 2018 International Design Engineering Technical Conf. and Computers and Information in Engineering Conf.; 2018 Aug 26–29; Quebec, Canada.