



## Motion paths finding for multi-degree-of-freedom mechanisms

Yang Li<sup>a,b</sup>

<sup>a</sup> The Institute of Technological Sciences, Wuhan University, Wuhan 430072, PR China

<sup>b</sup> Graduate Aerospace Laboratories, California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, United States



### ARTICLE INFO

#### Keywords:

Multi-degree-of-freedom  
Mechanisms  
Motion path-finding  
Origami  
Shooting method

### ABSTRACT

Multiple degree-of-freedom (DOF) mechanisms can provide more flexible reconfigurations than 1 DOF ones but are more complex to analyze. One difficulty is the existence of an unlimited number of valid motion directions at each configuration point. In this paper, a shooting method that is defined by customized target functions is introduced to help determine a specific motion direction at each configuration point. As a result, an incremental path-finding simulation can be carried out by following a unique motion direction at each step. The ability to arbitrarily define the target function allows a flexible and desirable exploration of the motion space, and it can find different motion paths that lead to the target state. Two examples are provided, in which different motion paths for deploying a 5-bar linkage and folding a Resch origami pattern are generated.

## 1. Introduction

### 1.1. Mechanisms and morphing structures

Mechanisms [1] consisting of rigid members and hinges are commonly used as skeletons, where additional elastic components and actuators can be mounted on top of them, for morphing structures. These structures can be reconfigurable [2,3], which can provide morphing platforms that need several distinct service configurations, such as for reconfigurable antennas [4,5] and morphing wings of aircrafts [6,7]. As a result, more integrated and compound-functional devices can be developed. Morphing structures can also be packageable in transportation and deployable in service [8,9], which can be used for space solar panels in aerospace applications [10] and stent grafts in medical applications [11]. Another type of applications is in robotics [12,13] that uses morphing structures to achieve certain motions.

### 1.2. Multi-DOF mechanisms

Most research into mechanism (the skeleton of a morphing structure) analysis focuses on one degree-of-freedom (DOF) mechanisms, such as four-bar linkages [14] and Miura-ori [15], which have analytical expressions of the kinematics and are easy to control in practice. However, their flexibility of motion is correspondingly limited by the kinematic simplicity. Multi-DOF mechanisms can morph more flexibly to many different configurations and are attracting more attentions recently [3,16–21]. When a multi-DOF mechanism is employed (due to the necessity), there are two practical ways to control/actuate it:

- Mount sufficient number (> DOF) of actuators on it (such as in [22]), and then coordinate those actuators in a kinematically compatible way, so that (1) no undesired self-stress and no actuation frustration are caused and (2) desired reconfiguration is reasonably achieved. This needs the knowledge of kinematic paths, and that is what this paper tries to achieve.
- Transform the multi-DOF mechanism into an elastic structure by introducing elastic components which correspondingly introduce an energy landscape. Design that energy landscape properly so that it helps to guide the motion and then requires fewer actuators to provide the desired reconfiguration. This method is not considered in this paper and is discussed in Li and Pellegrino [23].

This paper mainly investigates different kinematic paths connecting two known configurations, which can then be used to assess the performance of the mechanism and provide guidance for the morphing-actuation control.

### 1.3. Analyzing multi-DOF mechanisms

The literature on mechanism analysis is extensive. The motion of 1 DOF mechanisms has complex closed-form expressions [14,24,25]. It becomes harder or impossible to analytically track the motion, which consists of solutions that satisfy kinematic constraints, of complex mechanisms that have more than one independent kinematic variable. As a consequence, numerical simulation is typically used to study multi-DOF mechanisms. There are two purposes/types of numerical simulations:

E-mail address: [yangli.structures@gmail.com](mailto:yangli.structures@gmail.com)

Nomenclature	
$\mathbf{C}^b$	$n_b \times 3n$ compatibility matrix for all bars
$\mathbf{C}_k^b$	$1 \times 3n$ compatibility matrix for bar $k$
$\mathbf{C}^h$	$(n_{rh} + n_{fh}) \times 3n$ compatibility matrix for all hinges
$\mathbf{C}_r^h$	$1 \times 3n$ compatibility matrix for hinge $r$
$\mathbf{C}^{ch}$	$n_{rh} \times 3n$ compatibility matrix for all rotational hinges
$\mathbf{C}^{fh}$	$n_{fh} \times 3n$ compatibility matrix for all fixed hinges
$\mathbf{C}^c$	$n_c \times 3n$ compatibility matrix for all nodes subject to special boundary constraints
$\mathbf{C}^{ch}$	$n_{ch} \times 3n$ compatibility matrix for all activated local contact
$\mathbf{d}_t^i$	the target direction at configuration $i$
$\hat{\mathbf{d}}_m^i$	unnormalized first-order compatible displacements at configuration $i$
$\mathbf{d}_m^i$	normalized first-order compatible displacements at configuration $i$
$\mathbf{d}_c^{i,j}$	correction displacements for configuration $i$ and correction step $j$
$\mathbf{d}_c^i$	sum of all correction displacements for configuration $i$
$\mathbf{e}_b$	$n_b$ -dimensional vector of first-order extensions of all bars
$\mathbf{e}_b^{i,j}$	$n_b$ -dimensional vector of deformations of all bars at configuration $i$ and iteration step $j$
$\mathbf{e}_b^k$	first-order extension of bar $k$
$\mathbf{e}_c$	$n_c$ -dimensional vector of first-order displacements in the boundary-constrained directions of all nodes that subject to the corresponding constraint
$\mathbf{e}_c^k$	first-order displacement in the boundary-constrained direction of node $k$
$\mathbf{f}$	external nodal forces
$\mathbf{K}$	$n_h \times n_h$ diagonal matrix contains torsional springs stiffnesses
$m$	the number of inextensional mechanisms
$\mathbf{m}, \mathbf{n}_1, \mathbf{n}_2$	normal vectors for the hinge element
$n, n_b, n_h, n_{rh}, n_{fh}, n_c, n_{ch}$	the number of unconstrained nodes, the number of pin-jointed bars, the number of hinges, the number of rotational hinges, the number of fixed hinges, the number of nodes subjecting to special boundary constraints, the number of active local contacts
$\mathbf{U}$	$(n_b + n_{fh} + n_c + n_{ch}) \times (n_b + n_{fh} + n_c + n_{ch})$ matrix of left singular vectors for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{U}_{r_C}$	$(n_b + n_{fh} + n_c + n_{ch}) \times r_C$ matrix of left singular vectors corresponding to the non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{U}^h$	$(n_b + n_{fh} + n_c + n_{rh}) \times (n_b + n_{fh} + n_c + n_{rh})$ matrix of left singular vectors for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{U}_{r_h}^h$	$(n_b + n_{fh} + n_c + n_{rh}) \times r_h$ matrix of left singular vectors corresponding to the non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{V}$	$(n_b + n_{fh} + n_c + n_{ch}) \times 3n$ matrix of the singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{V}_{r_C}$	$r_C \times r_C$ diagonal matrix of the non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{V}^h$	$(n_b + n_{fh} + n_c + n_{rh}) \times 3n$ matrix of the singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{V}_{r_h}^h$	$r_h \times r_h$ diagonal matrix of the non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{W}$	$3n \times 3n$ matrix of the right singular vectors for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{W}_{r_C}$	$3n \times r_C$ matrix of the right singular vectors corresponding to non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{W}_m$	$3n \times m$ matrix of the right singular vectors corresponding to zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{ch} \mathbf{T}]^T$
$\mathbf{W}^h$	$3n \times 3n$ matrix of the right singular vectors for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{W}_{r_h}^h$	$3n \times r_h$ matrix of the right singular vectors corresponding to non-zero singular values for matrix $[\mathbf{C}^b \mathbf{T} \mathbf{C}^{fh} \mathbf{T} \mathbf{C}^c \mathbf{T} \mathbf{C}^{rh} \mathbf{T}]^T$
$\mathbf{x}^i, \mathbf{x}_t$	$3n$ -dimensional vector of the nodal coordinates of configuration $i$ and the target configuration
$\mathbf{x}^{i,j}$	$3n$ -dimensional vector of the nodal coordinates of configuration $i$ to which the motion displacements $\mathbf{d}_m^i$ and $j$ times correction displacements have been applied
$\alpha$	increment step size
$\epsilon_c$	tolerance of violation of compatibility constraints
$\epsilon_t$	tolerance of reaching the vicinity of the target configuration
$\theta$	$n_h$ -dimensional vector of first-order rotation for all hinges
$\theta_r$	first-order rotation for hinge $r$
$\theta_{rh}$	$n_{rh}$ -dimensional vector of first-order rotation for all rotational hinges
$\theta_{fh}$	$n_{fh}$ -dimensional vector of first-order rotation for all fixed hinges
$\tau$	$n_h$ -dimensional vector of internal moments of all torsional springs
$\phi_t, \phi^i$	fold angles of all rotational hinges for the target configuration, fold angles of all rotational hinges for configuration $i$
$\phi_0$	$n_h$ -dimensional vector of rest angles of all torsional springs

$\phi_{fh}^{i,j}$	$n_{fh}$ -dimensional vector of deformation for all fixed hinges for configuration $\mathbf{x}^{i,j}$
-------------------	---

- *Predictive simulations* that assign both the initial configuration and the actuation of the mechanism, and then compute for the end configuration and its motion process.
- *Path-finding simulations* that assign both of the initial and the end configurations of the mechanism, and then compute for the actuation needed and its motion process.

To run a simulation, there are two typical ways of parameterizing/modeling a mechanism:

- *The plate model* [26] that regards fold angles as variables and the closure conditions for each linkage loop and origami vertex as constraints. Mechanisms parameterized by fold angles have fewer variables and simpler mathematical behaviors, i.e., smoother kinematic manifolds (constraint space), than being parameterized by nodal coordinates.
- *The truss model* [26] that regards nodal coordinates as variables and connections between nodes as constraints. It has been widely employed in computational mechanics, such as in the matrix stiffness method and finite element method. Recently, hinge element has been proposed in Liu and Paulino [27] which provides the relationship between nodal coordinates and fold angles. It enables the truss model to work as the plate model, and vice versa.

This paper mainly focuses on using the truss model (including hinge elements) to run path-finding simulations, in which the initial and target configurations are given and then the different kinematic paths (or different displacement/rotation-controlled actuation schemes) are computed. The same framework can be used also for predictive simulations where stiffness and external forces can be added to the multi-DOF mechanism. Some of most related literatures are introduced below.

#### 1.4. Literature review

Tachi [28] developed a numerical method based on the plate model that can run both predictive and path-finding (by assigning target fold angles) simulations for complex origami mechanisms. He linearized the constraints, worked out first-order compatible motion modes, and correspondingly developed an incremental simulation scheme that can handle the analysis of complex multi-DOF mechanisms, as demonstrated by Tachi [29]. Xi and Lien [30,31] used plate model to run path-finding simulations for multi-DOF origami mechanisms. They used linear sampling to generate many intermediate incompatible configurations (constraints are violated) between the initial and target configurations. Respective nearby compatible configurations (constraints are satisfied) could then be found via random search in order for a valid path to be identified. However, correcting the incompatibility (violation of constraints) by random search results in slow convergence in high-dimensional space, and the linear sampling method might produce segmented paths (instead of a continuous one) in a complex motion space due to nonlinearity and bifurcation. The plate model is also used to run path-finding simulations for multi-DOF linkages [20,21] in the field of robotics, where random search and variable controlling/assigning are used. They are less practical for more complex mechanisms due to the high dimension of constraint space.

The truss model is employed in Overvelde et al. [3], [16], Liu and Paulino [27], Kumar and Pellegrino [32], Zhang et al. [33] to run predictive simulations. The truss model has a better correlation with computational mechanics, and it is ready to compute states of self-stress, equilibrium, and so on. It is also convenient to apply nodal forces or nodal displacements which are the most common actuation methods. Furthermore, stiffness can be assigned to hinges (using hinge elements) of multi-

DOF mechanisms and transfer them into elastic structures. Through applying external nodal loads or displacements to an elastic structure, a determined motion path can be generated based on energy minimization, as demonstrated by Overvelde et al. [3], [16], Liu and Paulino [27], Zhang et al. [33]. Similarly, finite element analysis using commercial software of the resultant elastic structure (created by introducing stiffness into hinges) can be carried out [18]. The introduction of energy (hinge stiffness) helps to specify a motion direction for each increment (following the energy gradient) but also limits the flexibility of exploring possible paths in more arbitrary ways.

To summarize, a toolbox that (1) can run path-finding simulations and provide many different paths for complex multi-DOF mechanisms and (2) is based on the truss model that has a better correlation with computational mechanics is missing in the literature, and this toolbox is provided in this paper.

#### 1.5. The paper layout

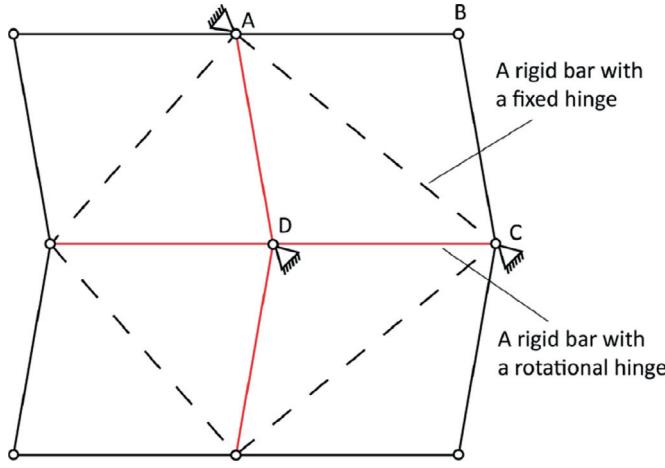
This paper uses the truss model (with hinge elements) and the shooting method based on the incrementation scheme provided by Kumar and Pellegrino [32] which is expanded for path-finding simulations of multi-DOF mechanisms. The main advancement comes in providing arbitrary ways to specify a compatible motion direction in each increment. These different specifications are then used to explore the motion space in a desirable way, for example, by finding paths with different features that lead to the same target configuration.

The layout of the paper is as follows. Section 2 gives the kinematic constraints that can compute the deformation of mechanisms and the first-order compatibility. Section 3 provides simulation algorithms to explore the motion space (defined by the kinematic constraints) in several different ways. Two examples are provided in Section 4, which contains a simple example of deploying a five-bar linkage and a complex example of (un)folding a Resch-ori pattern by following several different paths. Section 5 concludes the paper. A nomenclature list is also provided, followed by an Appendix with demonstration codes, proof of convergence, and some matrices of the example in Section 4. Supplementary material that includes motion movies of the example and simulation codes is provided.

## 2. Constraints based on the truss model: deformation computation and first-order compatibility

This section derives kinematic constraints of mechanisms. Consider a mechanism consisting of  $n_b$  pin-jointed rigid bars (Section 2.1),  $n_{rh}$  rotational hinges (Section 2.2),  $n_{fh}$  fixed hinges (Section 2.2), and  $n_c$  boundary nodes (Section 2.3) that are subjected to special boundary constraints. Note that any three-dimensional bar-and-plate structure can be modeled in terms of pin-jointed bars, by modeling the bars directly with pin-jointed bars, and subdividing the plates into triangles. Then, the edges of each triangle are modeled with pin-jointed bars and the planarity of the triangles that belong to the same plate is modeled by means of a no-rotation constraint, i.e., fixed hinges, along the common edges. The number of unconstrained nodes of the structure is  $n$ , hence, the total number of degrees of freedom is  $3n$  (before considering any kinematic constraints). An example of a unit of Miura-ori is given in Fig. 1, where  $n_b = 16$ ,  $n_{rh} = 4$ ,  $n_{fh} = 4$ ,  $n = 6$ , and  $n_c = 0$ . The coordinates of the initial configuration are  $\mathbf{x}^0$ , and the coordinates of the current configuration are  $\mathbf{x}^{i,j}$ , having experienced  $i$  prediction steps and  $j$  correction steps (explained in Section 3).

A deformation computation computes the exact geometric difference between the coordinates  $\mathbf{x}^0$  and  $\mathbf{x}^{i,j}$ . For instance, the change of distance between two nodes and the variation of fold angle between two triangles that share a common edge. First-order compatibility provides a relationship between the first-order deformation (extension or rotation) and nodal displacements.



**Fig. 1.** A unit of Miura-ori that is represented as a bar-and-plate structure where A, C, and D are pinned to the foundation. Each line segment represents a pin-jointed bar. The dash line indicates an additional non-rotational constraint is applied to the bar. The red line implies that the corresponding dihedral angle is free to change. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 2.1. Pin-jointed bars

### 2.1.1. Deformation computation

The lengths of all bars with  $\mathbf{x}^0$  can be computed using the distance formula and are considered to be the original bar lengths. Similarly, the lengths of the bars with  $\mathbf{x}^{i,j}$  can be achieved, and their differences from the original bar lengths constitute the deformation, which is denoted  $\mathbf{e}_b^{i,j}$ .

### 2.1.2. First-order compatibility

Consider a general bar  $k$  of a pin-jointed structure that connects node  $i$  to node  $j$ , as shown in Fig. 2. When small displacements  $\mathbf{d}_k = [u_k^i \ v_k^i \ u_k^j \ v_k^j]^T$  are applied, their first-order contribution to the bar extension  $e_b^k$  can be expressed as

$$-\cos \alpha_k u_k^i - \sin \alpha_k v_k^i + \cos \alpha_k u_k^j + \sin \alpha_k v_k^j = e_b^k. \quad (1)$$

Since  $\cos \alpha_k$  and  $\sin \alpha_k$  can be expressed through the coordinates of nodes  $i$  and  $j$ , Eq. (1) can be rewritten as

$$\frac{x_i - x_j}{L_k} u_k^i + \frac{y_i - y_j}{L_k} v_k^i + \frac{x_j - x_i}{L_k} u_k^j + \frac{y_j - y_i}{L_k} v_k^j = e_b^k, \quad (2)$$

which can be extended easily to 3 dimensions.

Putting together the compatibility equations of all bars, they can be written in a matrix form as

$$\mathbf{C}^b \mathbf{d} = \mathbf{e}_b, \quad (3)$$

where  $\mathbf{C}^b$  is a  $n_b \times 3n$  matrix in which each row can have up to 6 non-zero components,  $\mathbf{d}$  is the nodal displacement vector with  $3n$  components, and  $\mathbf{e}_b$  is the first-order extension vector of bars with  $n_b$  components. For rigid bars,  $\mathbf{e}_b$  is required to be  $\mathbf{0}$ . The corresponding solution of  $\mathbf{d}$ , which is in the null space of  $\mathbf{C}^b$ , gives the first-order compatible displacements that provide no first-order bar extension. This derivation is also provided in Pellegrino and Calladine [34]. A Matlab routine for generating  $\mathbf{C}^b$  (with given nodal coordinates, simple boundary conditions, and bar connections) is presented in Appendix A.1.

## 2.2. Hinges

Next, consider a general hinge connection,  $r$ , between the triangles connecting nodes  $i, j, k$ , and  $l$ . There are two types of hinges that we are interested in, i.e., rotational hinges and fixed hinges. Equations provided in this subsection are for the general case of two triangles connected by a hinge at an initial fold angle  $\phi$ , as shown in Fig. 3.

### 2.2.1. Deformation computation

Based on the given coordinates  $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l$ , define  $\mathbf{r}_{ki}, \mathbf{r}_{ji}, \mathbf{r}_{li}, \mathbf{r}_{kj}$ , and  $\mathbf{r}_{lj}$  as in Fig. 3. For example, let  $\mathbf{r}_{ki} = \mathbf{x}_k - \mathbf{x}_i$ . Then the fold angle

$$\phi = \begin{cases} \hat{\phi} & \text{if } a \geq 0, b \geq 0 \text{ (obtuse valley fold)} \\ -\hat{\phi} & \text{if } a \geq 0, b < 0 \text{ (obtuse mountain fold)} \\ \pi - \hat{\phi} & \text{if } a < 0, b \geq 0 \text{ (acute valley fold)} \\ -\pi + \hat{\phi} & \text{if } a < 0, b < 0 \text{ (acute mountain fold)} \end{cases} \quad (4)$$

where

$$\hat{\phi} = \arctan\left(\left|\frac{b}{a}\right|\right), \quad (5)$$

$$a = \mathbf{n}_1 \cdot \mathbf{n}_2, \quad b = \mathbf{m} \cdot \mathbf{n}_2, \quad (6)$$

and

$$\mathbf{n}_1 = \frac{\mathbf{r}_{ki} \times \mathbf{r}_{ji}}{\|\mathbf{r}_{ki} \times \mathbf{r}_{ji}\|}, \quad \mathbf{n}_2 = \frac{\mathbf{r}_{ji} \times \mathbf{r}_{li}}{\|\mathbf{r}_{ji} \times \mathbf{r}_{li}\|}, \quad \mathbf{m} = \frac{\mathbf{r}_{ji} \times \mathbf{n}_1}{\|\mathbf{r}_{ji} \times \mathbf{n}_1\|}. \quad (7)$$

Note that  $\phi = 0$  when the triangles are coplanar. An obtuse-acute fold refers to the dihedral angle between two triangles, and the mountain-valley fold is defined according to the perspective shown in Fig. 3, where  $i, l, j$ , and  $k$  are arranged in clockwise order. A Matlab routine for computing  $\phi$  for one hinge is given in Appendix A.2. For fixed hinges, the original fold angles and current fold angles can be computed based on coordinates  $\mathbf{x}^0$  and  $\mathbf{x}^{i,j}$ , respectively. The current values minus the original values gives the deformation  $\phi_{fh}^{i,j}$ .

### 2.2.2. First-order compatibility

Now,  $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ , and  $\mathbf{x}_l$  are the four nodal coordinates that define the hinge, while  $d_1, d_2, l_1, l_2, l_3$ , and  $l_4$  describe the relative position of the four nodes, as shown in Fig. 4. The linearized relationship between the nodal displacements  $\mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_k, \mathbf{d}_l$ , and the first-order hinge rotation  $\theta_r$  can be expressed as

$$-\left(\frac{\mathbf{n}_1 l_1}{d_1 L} + \frac{\mathbf{n}_2 l_3}{d_2 L}\right) \cdot \mathbf{d}_i - \left(\frac{\mathbf{n}_1 l_2}{d_1 L} + \frac{\mathbf{n}_2 l_4}{d_2 L}\right) \cdot \mathbf{d}_j + \frac{\mathbf{n}_1}{d_1} \cdot \mathbf{d}_k + \frac{\mathbf{n}_2}{d_2} \cdot \mathbf{d}_l = \theta_r, \quad (8)$$

by summing up the first-order rotation contributions from  $\mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_k, \mathbf{d}_l$  respectively, where

$$L = \|\mathbf{r}_{ji}\|, \quad d_1 = \|\frac{\mathbf{r}_{ki} \times \mathbf{r}_{ji}}{\|\mathbf{r}_{ki}\| \|\mathbf{r}_{ji}\|}\| \cdot \|\mathbf{r}_{ki}\|, \quad d_2 = \|\frac{\mathbf{r}_{ji} \times \mathbf{r}_{li}}{\|\mathbf{r}_{ji}\| \|\mathbf{r}_{li}\|}\| \cdot \|\mathbf{r}_{li}\|, \quad (9)$$

and

$$l_1 = \sqrt{\mathbf{r}_{kj}^2 - d_1^2}, \quad l_2 = \sqrt{\mathbf{r}_{ki}^2 - d_1^2}, \\ l_3 = \sqrt{\mathbf{r}_{lj}^2 - d_2^2}, \quad l_4 = \sqrt{\mathbf{r}_{li}^2 - d_2^2}. \quad (10)$$

Note that  $\theta_r$  has a positive sign when the fold angle  $\phi$  is increasing. Eq. (8) can be expressed in matrix form as

$$\mathbf{C}_r^h \mathbf{d} = \theta_r. \quad (11)$$

A Matlab routine for generating  $\mathbf{C}_r^h$  is given in Appendix A.3. Putting the compatibility relationships of all hinges together in matrix form gives

$$\mathbf{C}^h \mathbf{d} = \theta, \quad (12)$$

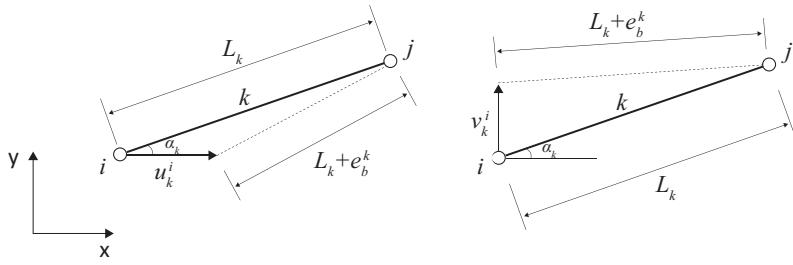
where  $\mathbf{C}^h$  has size  $(n_{rh} + n_{fh}) \times 3n$  in which each row can have up to 12 non-zero components. The same relationship can be derived in a different way by Liu and Paulino [27].

Since there are two types of hinges that are of interest in this paper, Eq. (12) can be divided into two as

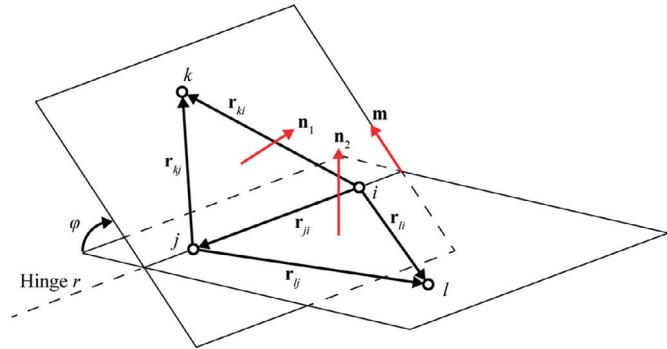
$$\mathbf{C}^{rh} \mathbf{d} = \theta_{rh} \quad (13)$$

for rotational hinges where  $\mathbf{C}^{rh}$  has size  $n_{rh} \times 3n$ , and

$$\mathbf{C}^{fh} \mathbf{d} = \theta_{fh} \quad (14)$$



**Fig. 2.** Compatibility of bar  $k$  where the solid line indicates the undeformed bar with length  $L_k$  and the dashed line represents the deformed bar with length  $(L_k + e_b^k)$  due to displacements  $u_k^i$  and  $v_k^i$  respectively.



**Fig. 3.** A general hinge connection  $r$  defined by nodes  $i, j, k$ , and  $l$  where  $n_1$  and  $n_2$  are normal vectors of the respective triangle plate. All information noted in the figure can be computed based on coordinates of  $i, j, k$ , and  $l$ .

for fixed hinges, where  $\mathbf{C}^{fh}$  has size  $n_{fh} \times 3n$ . Note that  $\theta_{fh}$  is required to be  $\mathbf{0}$  for first-order compatible displacements. Given the nodal coordinates and bar connections, the nodes that are associated with each hinge can be identified using the codes given in [Appendix A.4](#). The fold angles and compatibility matrix for all hinges can be computed using the codes given in [Appendix A.5](#).

### 2.3. Boundary conditions

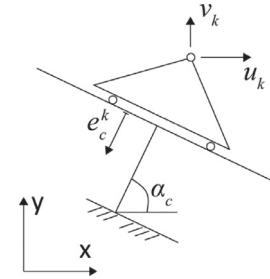
Different boundary conditions and special constraints can be applied to the nodes. Here, a sliding boundary condition for node  $k$  in the  $xy$  plane is demonstrated and shown in [Fig. 5](#).

Computing the distances of boundary nodes of  $x^{i,j}$  from the sliding plane [35] gives deformation  $e_c^{i,j}$ .

First-order compatibility gives

$$-\cos \alpha_c u_k - \sin \alpha_c v_k = e_c^k \quad (15)$$

where  $e_c^k$  is the displacement perpendicular to the sliding boundary, and  $\alpha_c$  is the angle between the reaction bar and the horizontal plane, which,



**Fig. 5.** The sliding boundary constraint where the direction that is perpendicular to  $e_c^k$  is free.

in this case, is independent from the coordinates. Putting together the compatibility relationship of all nodes ( $n_c$  of them) that are subject to the sliding boundary constraints gives

$$\mathbf{C}^c \mathbf{d} = \mathbf{e}_c, \quad (16)$$

where  $\mathbf{C}^c$  has size  $n_c \times 3n$ , and  $\mathbf{e}_c$  is required to be  $\mathbf{0}$  for first-order compatible displacements.

### 2.4. Activated local contact

Consider that as the fold angle of the rotational hinge  $r$  gets close to  $\pi$ , i.e., larger than  $\pi - \epsilon_{ch}$ , where  $\epsilon_{ch}$  is the contact detection tolerance, it activates the local contact constraint to enforce

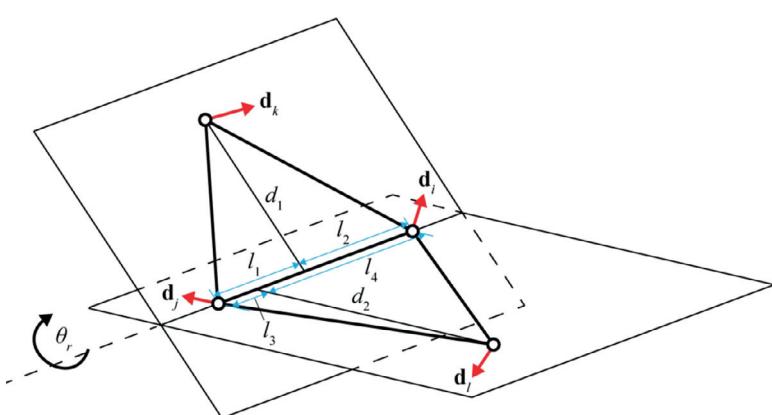
$$\mathbf{C}_r^h \mathbf{d} \leq 0, \quad (17)$$

where  $\mathbf{C}_r^h$  is based on [Eq. \(8\)](#) and of size  $1 \times 3n$ , and only entries corresponding to nodes associated with hinge  $r$  can have non-zero components. Similarly, if the fold angle of the rotational hinge  $r$  is close to  $-\pi$ , i.e., smaller than  $-\pi + \epsilon_{ch}$ , the constraint is

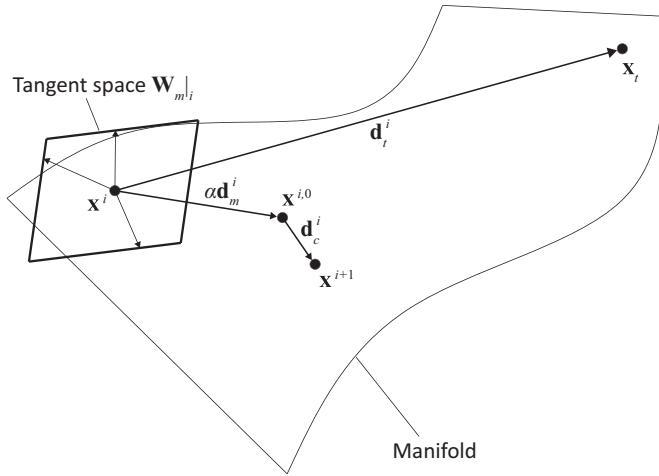
$$(-\mathbf{C}_r^h) \mathbf{d} \leq 0. \quad (18)$$

Putting the constraints for all activated contact hinges together gives

$$\mathbf{C}^{ch} \mathbf{d} \leq 0, \quad (19)$$



**Fig. 4.** Geometric relationship of a hinge.  $\mathbf{d}_i$ ,  $\mathbf{d}_j$ ,  $\mathbf{d}_k$ , and  $\mathbf{d}_l$  are infinitesimal nodal displacements.  $d_1, d_2, l_1, l_2, l_3$ , and  $l_4$  define the relative position of nodes  $i, j, k$ , and  $l$ .  $\theta_r$  is the infinitesimal rotation due to nodal displacements.



**Fig. 6.** The algorithm to find  $\mathbf{x}^{i+1}$  (the next configuration point) from  $\mathbf{x}^i$  (the current configuration point) in the configuration space parameterized by nodal coordinates.

where  $\mathbf{C}^h$  has size  $n_{ch} \times 3n$ , and  $n_{ch}$  is the number of activated local contacts.

When the step size,  $\alpha$ , is large (comparing to  $\epsilon_{ch}$ ), contact violation can occur and is detected by tracking values of all fold angles computed from Eq. (4) in each step. When there is a sudden flip of a fold angle from close to  $\pi$  to close to  $-\pi$  or vice versa, the corresponding hinge is identified as violating the contact constraint. The amount of contact violation can be worked out by comparing the current fold angle to  $\pi - \epsilon_{ch}$  or  $-\pi + \epsilon_{ch}$  whichever has the same sign as the fold angle in the previous step (contact free). Putting the amounts of violation for all contact-violated hinges together for configuration  $\mathbf{x}^{i,j}$  gives  $\phi_{ch}^{i,j}$ .

### 3. Simulation algorithms based on constraints

Employing the constraints given in Section 2, this section provides algorithms for computing kinematic paths that are represented by a series of configuration points and connect a given configuration to a pre-scribed/target state. The simulation uses a prediction-correction incrementation method as in Kumar and Pellegrino [32].

The process can be stated in a visual way. Within the configuration space, consider a given point  $\mathbf{x}^i$  that lives in the manifold where the compatibility constraints are satisfied, as shown in Fig. 6. Define a target point  $\mathbf{x}_t$  and a corresponding target direction  $\mathbf{d}_t^i$ . Project  $\mathbf{d}_t^i$  into the tangent space  $\mathbf{W}_{m|_i}$  of the manifold that goes through  $\mathbf{x}^i$ .  $\mathbf{W}_{m|_i}$  has  $m$  dimensions for a  $m$ -DOF mechanism and can be computed based on first-order compatibility constraints. As a result, a first-order compatible displacement  $\mathbf{d}_m^i$  is produced. Multiply  $\mathbf{d}_m^i$  by a small coefficient  $\alpha$  and apply the product to  $\mathbf{x}^i$ , it produces  $\mathbf{x}^{i,0}$  that is slightly outside the manifold due to higher-order (higher than the first-order) residuals. Correction displacements  $\mathbf{d}_c^i$  are then applied to  $\mathbf{x}^{i,0}$ , resulting in a new current configuration  $\mathbf{x}^{i+1}$  that satisfies the constraints (within the manifold) and gets closer to  $\mathbf{x}_t$ . Iterating this process until the current configuration reaches the vicinity of  $\mathbf{x}_t$  leaves a valid path consisting of [...]  $\mathbf{x}^i$  [...]. To summarize, one prediction-correction incrementation step can be divided into four steps:

- Step 1: From the current configuration  $\mathbf{x}^i$ , define the target nodal displacements, such as  $(\mathbf{x}_t - \mathbf{x}^i)$  in Section 3.1, or target deformation, such as  $(\boldsymbol{\phi}_t - \boldsymbol{\phi}^i)$  in Section 3.3.
- Step 2: Compute the corresponding kinematically compatible displacements  $\mathbf{d}_m^i$  based on Section 3.2 for target nodal displacements, or Section 3.3 for target deformation.
- Step 3: Move along  $\mathbf{d}_m^i$  with a small step  $\alpha$  as in Section 3.4.

Step 4: Correct the higher order violation of constraints by applying  $\mathbf{d}_c^i$ , and it gives the next configuration  $\mathbf{x}^{i+1}$ , as shown in Section 3.5.

Repeat this prediction-correction incrementation step until the current configuration satisfies the customized target. Derivations of all the required information are given in following subsections.

#### 3.1. The target direction $\mathbf{d}_t^i$

For a multi-DOF mechanism, there are unlimited motion directions, and a preference direction, which is the target direction  $\mathbf{d}_t^i$  (for configuration  $i$ ) defined in this section, is needed to determine a specific motion direction at each increment. Several examples of prescribing the preference direction  $\mathbf{d}_t^i$  are given below.

##### 3.1.1. A target state described by the nodal coordinates

Based on a target configuration

The nodal coordinates of a target configuration  $\mathbf{x}_t$  are given. The task is to find a path that connects the current configuration  $\mathbf{x}^i$  with it. Preference can be assigned to getting closer to the target configuration in terms of nodal coordinates at each increment. The corresponding target direction is

$$\mathbf{d}_t^i = \mathbf{x}_t - \mathbf{x}^i. \quad (20)$$

Based on other features

It can be of interest to find a path that leads to a configuration where selected nodes are at certain locations. For instance, compute target displacements that morph the selected nodes of the current configuration into a target spherical surface with center at  $\mathbf{x}_s$  and radius  $r$ . For configuration  $i$  and node  $k$ ,

$$\hat{\mathbf{d}}_t^i|_k = \frac{(\mathbf{x}_k^i - \mathbf{x}_s) \cdot r}{\|\mathbf{x}_k^i - \mathbf{x}_s\|} + \mathbf{x}_s - \mathbf{x}_k^i. \quad (21)$$

Compiling the displacements of all selected nodes together gives  $\mathbf{d}_t^i$ , which moves the selected nodes of the mechanism towards the target spherical surface.

##### 3.1.2. Energy minimization analogy

With the assignment of elastic properties to hinges, an analysis can be done to find the motion path to a stable equilibrium configuration [27,33]. By providing the target direction in the direction of energy gradient and then projecting it into the tangent space of the kinematic manifold, a similar analysis can be carried out by following the resultant direction at each increment.

Consider the torsional springs that are put on all rotational hinges and have rest angles  $\boldsymbol{\phi}_0$  and stiffness  $\mathbf{K}$  (a diagonal matrix with the elastic spring stiffnesses along the main diagonal). For configuration  $i$ , the internal moments are

$$\boldsymbol{\tau}^i = \mathbf{K}(\boldsymbol{\phi}_0 - \boldsymbol{\phi}^i), \quad (22)$$

and external nodal loads  $\mathbf{f}$  are applied. The residual nodal loads (excluding the consideration of kinematics defined by rigid components) are

$$\mathbf{d}_t^i = (\mathbf{C}^h)_i^T \mathbf{K}(\boldsymbol{\phi}_0 - \boldsymbol{\phi}^i) + \mathbf{f}. \quad (23)$$

where  $(\mathbf{C}^h)_i^T$  is the equilibrium matrix [34]. The residual nodal loads  $\mathbf{d}_t^i$  give the energy's steepest decent direction.

#### 3.2. The first-order compatible displacement $\hat{\mathbf{d}}_m^i$ based on $\mathbf{d}_t^i$

As  $\mathbf{d}_t^i$  is not tangent to the kinematic manifold, it needs to be projected into the tangent space that passes through  $\mathbf{x}^i$ , as shown in Fig. 6. Doing so gives the first-order compatible displacement  $\hat{\mathbf{d}}_m^i$  at configuration  $i$ .

### 3.2.1. Without contact

When no local contact is activated, i.e., no fold angles are close to  $\pm \pi$  or no contact is considered where  $\mathbf{C}^{ch} = \emptyset$  and  $n_{ch} = 0$ , the compatibility equations are in the form of

$$\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{fh} \\ \mathbf{C}^c \end{bmatrix}_i \hat{\mathbf{d}}_m^i = \mathbf{0}. \quad (24)$$

The SVD of the compatibility matrix is computed as

$$\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{fh} \\ \mathbf{C}^c \end{bmatrix}_i = \mathbf{U}|_i \mathbf{V}|_i \mathbf{W}|_i^T. \quad (25)$$

and the rank of the compatibility matrix is then given by the number of nonzero singular values  $r_C$ . The right-singular-vector matrix  $\mathbf{W}|_i$  has size  $3n \times 3n$ . It is divided into two submatrices in the following manner

$$\mathbf{W}|_i = [\mathbf{W}_{r_C}|_i \mathbf{W}_m|_i], \quad (26)$$

where the subscripts denote the number of columns in the submatrices. The columns of  $\mathbf{W}_m|_i$  form an orthonormal set of zero-energy (to the first order) deformation modes for the mechanism, which is the tangent space of the kinematic manifold that passes through  $\mathbf{x}^i$ . As a result, the projection of  $\hat{\mathbf{d}}_m^i$  into the tangent space is

$$\hat{\mathbf{d}}_m^i = \mathbf{W}_m|_i (\mathbf{W}_m|_i)^T \hat{\mathbf{d}}_m^i. \quad (27)$$

### 3.2.2. With contact

When the local contact is activated, Eq. (24) gives the linear equality constraints, and the additional linear inequality constraints (Eq. (19)) are

$$\mathbf{C}^{ch}|_i \hat{\mathbf{d}}_m^i \leq \mathbf{0}. \quad (28)$$

A unique  $\hat{\mathbf{d}}_m^i$  can be obtained by minimizing  $(\hat{\mathbf{d}}_t^i - \hat{\mathbf{d}}_m^i)^2$  (equivalently minimizing  $(\hat{\mathbf{d}}_m^i)^T \hat{\mathbf{d}}_m^i - 2(\hat{\mathbf{d}}_t^i)^T \hat{\mathbf{d}}_m^i$ ) subject to the constraints stated above, which forms a quadratic programming problem.

### 3.3. The first-order compatible displacement $\hat{\mathbf{d}}_m^i$ based on target deformation and hinge rotation

It can be of interest to achieve target distances between some nodes, target fold angles, and so on. For example, consider the target rotation  $\phi_t - \phi^i$ , where  $\phi_t$  are the target fold angles. Then the corresponding compatible displacement satisfies

$$\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{fh} \\ \mathbf{C}^c \\ \mathbf{C}^{rh} \end{bmatrix}_i \hat{\mathbf{d}}_m^i = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \phi_t - \phi^i \end{bmatrix}. \quad (29)$$

When no contact is considered,  $\hat{\mathbf{d}}_m^i$  can be solved using the SVD:

$$\hat{\mathbf{d}}_m^i = \mathbf{W}_{r_h^i}|_i (\mathbf{V}_{r_h^i})^{-1}_i (\mathbf{U}_{r_h^i})_i^T \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \phi_t - \phi^i \end{bmatrix}, \quad (30)$$

where

$$\mathbf{U}^h|_i \mathbf{V}^h|_i (\mathbf{W}^h)_i^T = \begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{fh} \\ \mathbf{C}^c \\ \mathbf{C}^{rh} \end{bmatrix}_i. \quad (31)$$

The rank of the matrix on the right-hand side of Eq. (31) is  $r_h^i$ , and the subscript of  $r_h^i$  indicates the inclusion of components that correspond to non-zero singular values only [36].

When the contact is activated, a unique  $\hat{\mathbf{d}}_m^i$  can be achieved by minimizing  $\|\hat{\mathbf{d}}_m^i\|$  subject to Eqs. (29) and (28).

### 3.4. Moving along $\hat{\mathbf{d}}_m^i$

Then  $\mathbf{d}_m^i$  is given by normalizing  $\hat{\mathbf{d}}_m^i$  via

$$\mathbf{d}_m^i = \frac{\hat{\mathbf{d}}_m^i}{\|\hat{\mathbf{d}}_m^i\|}. \quad (32)$$

A new configuration point can be generated as

$$\mathbf{x}^{i,0} = \mathbf{x}^i + \alpha \mathbf{d}_m^i, \quad (33)$$

where  $\alpha$  is the step size, which needs to be significantly smaller than 1. Here,  $\mathbf{x}^{i,0}$  moves towards the target direction/deformation within the tangent space. It has residuals (violation of constraints) on the order of  $\alpha^2$  that need to be corrected.

### 3.5. The correction displacement $\mathbf{d}_c^{i,j}$ and converging to the target state

Corrections are needed to eliminate higher-order residuals in  $\mathbf{x}^{i,0}$  where the superscript 0 indicates that the number of applied correction steps is 0. The deformation or amount of violation of the bar lengths, fold angles of fixed hinges, boundary constraints, and contact constraints of  $\mathbf{x}^{i,0}$  can be computed as  $[\mathbf{e}_b^T \ \boldsymbol{\phi}_{fh}^T \ \mathbf{e}_c^T \ \boldsymbol{\phi}_{ch}^T]_{i,0}$ , as illustrated in Section 2. The SVD of the compatibility matrix for  $\mathbf{x}^{i,0}$  with rank of  $r_C^{i,0}$  gives

$$\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{fh} \\ \mathbf{C}^c \\ \mathbf{C}^{ch} \end{bmatrix}_{i,0} = \mathbf{U}|_{i,0} \mathbf{V}|_{i,0} (\mathbf{W})_{i,0}^T. \quad (34)$$

Note that  $\boldsymbol{\phi}_{ch}^{i,j}$  and  $\mathbf{C}^{ch}|_{i,j}$  contain only the rotational hinges for which contact constraint has been violated. Violation of contact constraints is identified by noticing the sudden flip in values of the fold angles (from close to  $\pi$  to close to  $-\pi$  or vice versa) compared to the fold angles of  $\mathbf{x}^i$  (the previous step). The correction displacements of the residuals are

$$\mathbf{d}_c^{i,0} = -\mathbf{W}_{r_C^{i,0}}|_{i,0} (\mathbf{V}_{r_C^{i,0}})_{i,0}^{-1} (\mathbf{U}_{r_C^{i,0}})_{i,0}^T \begin{bmatrix} \mathbf{e}_b \\ \boldsymbol{\phi}_{fh} \\ \mathbf{e}_c \\ \boldsymbol{\phi}_{ch} \end{bmatrix}_{i,0} \quad (35)$$

and

$$\mathbf{x}^{i,1} = \mathbf{x}^{i,0} + \mathbf{d}_c^{i,0}. \quad (36)$$

Iterate this process  $j$  times until

$$\left\| \begin{bmatrix} \mathbf{e}_b^T & \boldsymbol{\phi}_{fh}^T & \mathbf{e}_c^T & \boldsymbol{\phi}_{ch}^T \end{bmatrix}_{i,j} \right\| < \epsilon_c, \quad (37)$$

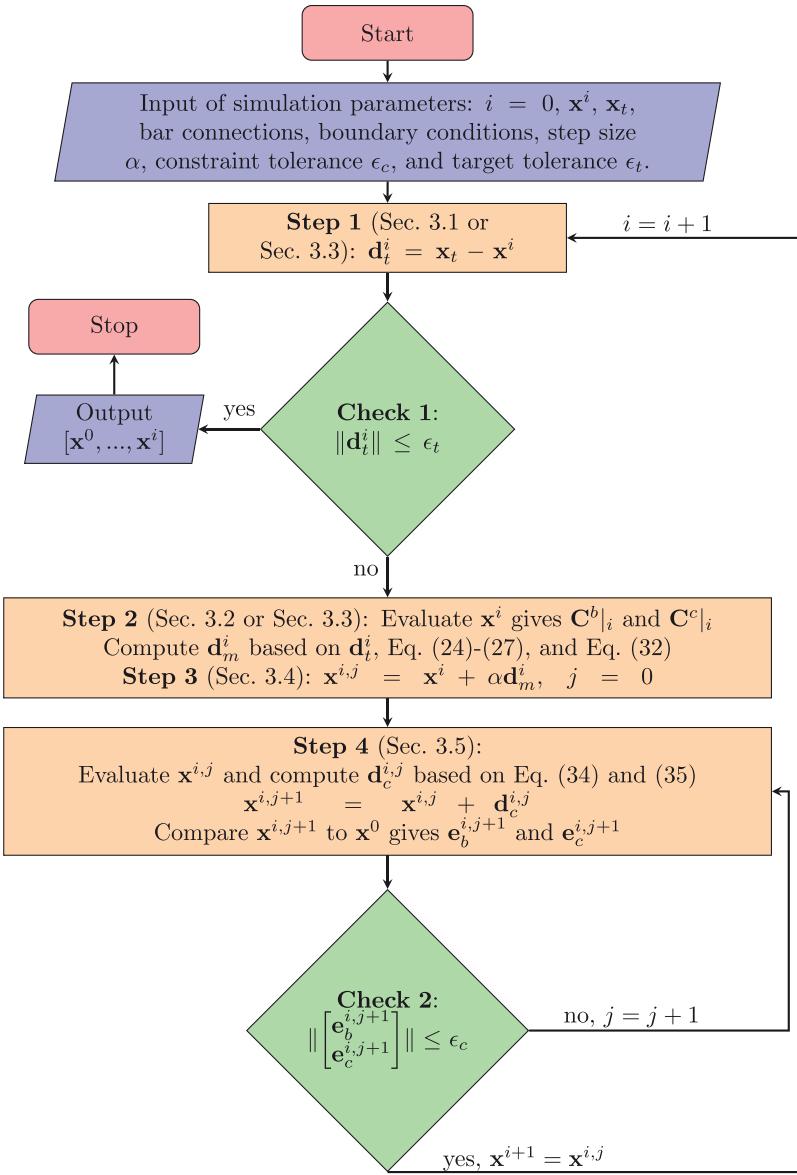
where  $\epsilon_c$  is the constraint tolerance. The convergence of this correction step is guaranteed if  $\alpha$  is sufficiently small, and this is proven in Appendix A.6. Then assign

$$\mathbf{x}^{i+1} = \mathbf{x}^{i,j} = \mathbf{x}^{i,0} + \mathbf{d}_c^i, \quad (38)$$

where  $\mathbf{d}_c^i = \sum_{k=0}^{j-1} \mathbf{d}_c^{i,k}$ , and  $\mathbf{x}^{i+1}$  satisfies the compatibility constraints (within the tolerance) and is closer to the target than  $\mathbf{x}^i$ . Repeat the whole process until the current configuration reaches the vicinity of the target. For instance, reaching the vicinity can be decided by

$$\|\mathbf{x}_t - \mathbf{x}^i\| < \epsilon_t, \quad (39)$$

where the current configuration index is assigned to  $i$  and  $\epsilon_t$  is the tolerance of reaching the target. However, converging to the target state is not guaranteed by the shooting method. This can be caused by (1) a highly nonlinear and non-convex kinematic manifold, (2) complex contact constraints, and (3) bifurcation paths. A possible solution is to (1) change the target function, (2) change the parametric space (of the target function), (3) swap the path-finding direction (such as in Section 4.2), and (4) introduce additional constraints to block unwanted kinematic branches. By employing these variations, convergence to the target state might be achieved.



**Fig. 7.** Flowchart of the simulation of a multi-DOF mechanism that has pin-jointed bars, sliding boundary conditions, no fixed hinge, and no contact constraints.

### 3.6. Flow chart of the analysis

As an example, a flowchart for simulating a simple multi-DOF mechanism that has pin-jointed bars, sliding boundary conditions, no fixed hinge, and no contact constraints is given in Fig. 7. The target coordinates (Eq. (20)) are used to generate the motion direction. Four steps of a prediction-correction incrementation step are noted in the chart. There are two convergence checks needed as shown in green diamond boxes as *Check 1* (Eq. (39)) and *Check 2* (Eq. (37)). *Check 1* is not guaranteed. *Check 2* is guaranteed if the step size  $\alpha$  is sufficiently small. This is proven mathematically in Appendix A.6.

## 4. Examples

### 4.1. A simple example: deployment paths of a five-bar linkage

#### 4.1.1. Introducing the problem

Consider a five-bar linkage (with 2 DOF) that consists of 5 nodes and 4 moving bars. Its initial configuration, which resembles its packaged state, is shown in Fig. 8(a), and the target configuration, which resembles the deployed state, is given in Fig. 8(b). The numbering of nodes

and bars are noted in the figure, and nodes 1 and 5 are pinned to the ground. Self-contact is not considered for simplicity. Nodal coordinates of the two configurations respectively are

$$\begin{bmatrix} -1 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} -1.0000 & 0 \\ -0.8264 & 0.9848 \\ 0 & 1.5480 \\ 0.8264 & 0.9848 \\ 1.0000 & 0 \end{bmatrix} \quad (40)$$

and can be rearranged as

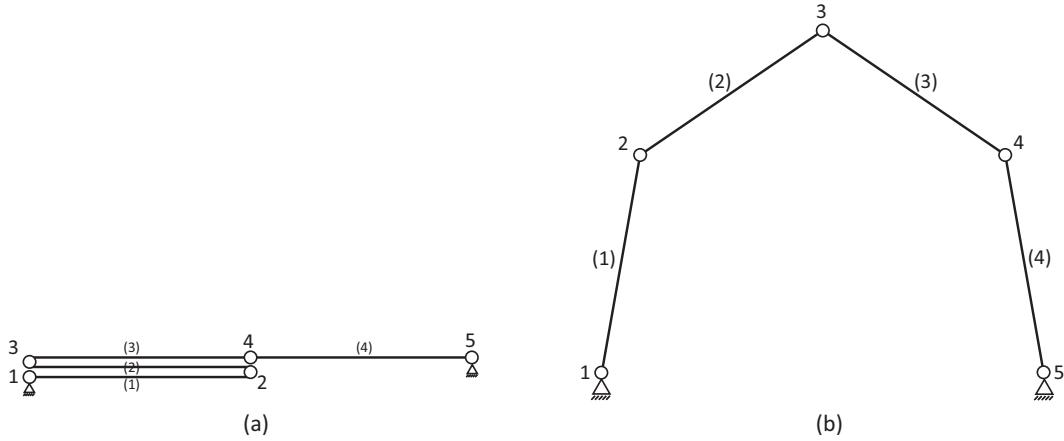
$$\mathbf{x}^0 = [-1 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 0 \ 1 \ 0]^T \quad (41)$$

and

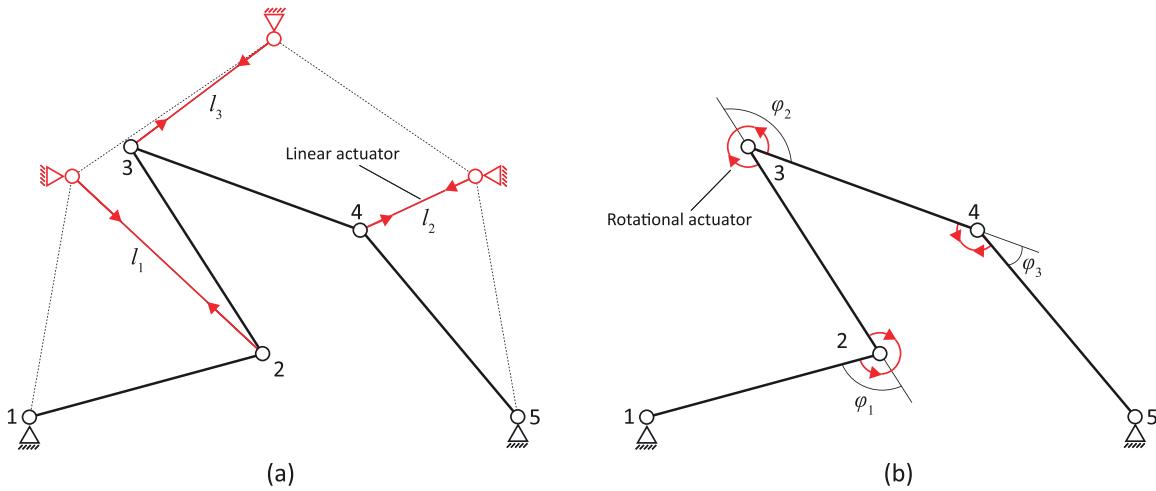
$$\mathbf{x}_t = [-1 \ 0 \ -0.8264 \ 0.9848 \ 0 \ 1.5480 \ 0.8264 \ 0.9848 \ 1 \ 0]^T. \quad (42)$$

The connections of nodes by means of rigid bars with length of 1 are as

$$[\text{Bars}] = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix}. \quad (43)$$



**Fig. 8.** Configurations of a 5-bar linkage that are (a) in the initial (packaged) state and (b) in the target (deployed) state.



**Fig. 9.** Different actuation schemes which are respectively (a) Situation A by three linear actuators and (b) Situation B by three rotational actuators.

Fold angles of hinges 2, 3, and 4 of the two configurations can be computed using Eq. (4) and respectively are

$$\phi^0 = \begin{bmatrix} -3.1416 \\ 3.1416 \\ 0 \end{bmatrix} \text{ and } \phi_i = \begin{bmatrix} -3.1289 \\ 3.1163 \\ 0.0232 \end{bmatrix}. \quad (44)$$

There are an infinite number of kinematic paths connecting the two configurations. As an illustrative example, this subsection generates two suitable paths based on two different user preferences. Consider two user preferences correspondingly in *Situation A* and *B*, as shown in Fig. 9. In Situation A (shown in Fig. 9(a)), the deployment is actuated by three displacement-controlled linear actuators, which can be by mechanical or electromechanical actuators. The requirement is to coordinate actuator lengths  $l_1, l_2$ , and  $l_3$  in a way that the deployment process is compatible and relatively fast. Generating the corresponding path employs *Scheme A* that is introduced in Section 4.1.2. In Situation B (Fig. 9(b)), the deployment is facilitated by the rotation-controlled actuation, which can be by stepper motors or servomotors, at hinges 2, 3, and 4. The correspondingly suitable path, which coordinates actuator angles  $\phi_1, \phi_2$ , and  $\phi_3$  in a way that the deployment process is compatible and relatively fast, is computed based on *Scheme B* in Section 4.1.3. Different kinematic paths are expected from the two schemes. Displacements of nodes 1 and 5 are removed in the following analysis.

#### 4.1.2. Scheme A: employing three linear actuators

It is equivalent to try to move nodes 2, 3, and 4 to target positions in a compatible and locally fastest way, which is described in Section 3.1.1.

Simulation parameters are assigned as

$$\alpha = 0.02, \epsilon_c = 10^{-12}, \text{ and } \epsilon_t = 0.02, \quad (45)$$

which are respectively the step size, the constraint tolerance, and the target tolerance. The choice of  $\alpha$  depends on two aspects: (1) the required resolution of the path, i.e., the number of configuration points that is needed; and (2) the nonlinearity of the kinematic manifold, for which  $\alpha$  needs to be small enough so that the new configuration point is close enough to the manifold and the correction step converges. The target direction of each incrementation is chosen according to Eq. (20), and for the initial step

$$\mathbf{d}_t^0 = \mathbf{x}_t - \mathbf{x}^0 = \begin{bmatrix} -0.8264 \\ 0.9848 \\ 1.0000 \\ 1.5480 \\ 0.8264 \\ 0.9848 \end{bmatrix}, \quad (46)$$

where nodes 1 and 5 are removed. The compatibility matrix of bars based on  $\mathbf{x}_0$  and Eq. (3) is

$$\mathbf{C}^b|_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}. \quad (47)$$

SVD of  $\mathbf{C}^b|_0$  based on Eqs. (25) and (26) gives

$$\mathbf{W}_m|_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (48)$$

which indicates that it has 3 inextensional modes instead of 2, because  $\mathbf{x}^0$  is at a bifurcation point. According to Eq. (27), projecting  $\mathbf{d}_t^0$  into  $\mathbf{W}_m|_0$  gives

$$\begin{aligned} \hat{\mathbf{d}}_m^0 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.8264 \\ 0.9848 \\ 1.0000 \\ 1.5480 \\ 0.8264 \\ 0.9848 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0.9848 \\ 0 \\ 1.5480 \\ 0 \\ 0.9848 \end{bmatrix}. \end{aligned} \quad (49)$$

According to Eqs. (32) and (33), normalizing  $\hat{\mathbf{d}}_m^0$  gives  $\mathbf{d}_m^0$ , and moving  $\mathbf{x}^0$  along  $\mathbf{d}_m^0$  by  $\alpha = 0.02$  gives the incrementation as

$$\mathbf{x}^{0,0} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + 0.02 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.4729 \\ 0 \\ 0.7434 \\ 0 \\ 0 \\ 0.4729 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1.0000 \\ 0 \\ 0 \\ 0.0095 \\ -1.0000 \\ 0.0149 \\ 0 \\ 0 \\ 0.0095 \\ 1.0000 \\ 0 \end{bmatrix}. \quad (50)$$

Note that nodes 1 and 5 are included for consistency. Second order extensions  $\mathbf{e}_b^{0,0}$  of bars at  $\mathbf{x}^{0,0}$  can be computed based on [Bars] (Eq. (43)) and Section 2.1.1 as

$$\mathbf{e}_b^{0,0} = 10^{-4} \times [0.4473 \quad 0.1463 \quad 0.1463 \quad 0.4473]^T. \quad (51)$$

The correction displacement based on Eq. (35) is

$$\mathbf{d}_c^{0,0} = -\mathbf{W}_{r_C}^{0,0}|_{0,0} (\mathbf{V}_{r_C}^{0,0})_{0,0}^{-1} (\mathbf{U}_{r_C}^{0,0})_{0,0}^T \mathbf{e}_b^{0,0} = \begin{bmatrix} 0.0000 \\ -0.0056 \\ 0.0001 \\ -0.0000 \\ 0.0000 \\ -0.0015 \end{bmatrix}, \quad (52)$$

where  $r_C^{0,0} = 4$ , and

$$\mathbf{W}_{r_C}^{0,0}|_{0,0} = \begin{bmatrix} -0.5000 & -0.7071 & 0.5000 & 0.0053 \\ 0.0005 & -0.0014 & 0.0100 & -0.9648 \\ 0.7071 & 0.0000 & 0.7070 & 0.0066 \\ -0.0038 & -0.0000 & -0.0038 & -0.0000 \\ -0.5000 & 0.7071 & 0.4999 & 0.0014 \\ 0.0033 & -0.0053 & -0.0062 & -0.2629 \end{bmatrix}, \quad (53)$$

$$(\mathbf{V}_{r_C}^{0,0})_{0,0}^{-1} = \begin{bmatrix} 0.5412 & 0 & 0 & 0 \\ 0 & 0.7071 & 0 & 0 \\ 0 & 0 & 1.3065 & 0 \\ 0 & 0 & 0 & 129.7971 \end{bmatrix}, \quad (54)$$

and

$$(\mathbf{U}_{r_C}^{0,0})_{0,0}^T = \begin{bmatrix} -0.27 & -0.65 & -0.65 & 0.27 \\ -0.5 & -0.5 & 0.5 & -0.5 \\ 0.65 & -0.27 & -0.27 & -0.65 \\ -0.5 & 0.5 & -0.5 & -0.5 \end{bmatrix}. \quad (55)$$

The configuration after one correction iteration is

$$\mathbf{x}^{0,1} = \mathbf{x}^{0,0} + \mathbf{d}_c^{0,0} = \begin{bmatrix} -1.0000 \\ 0 \\ 0.0000 \\ 0.0039 \\ -0.9999 \\ 0.0149 \\ 0.0000 \\ 0.0079 \\ 1.0000 \\ 0 \end{bmatrix}, \quad (56)$$

where nodes 1 and 5 are included. The corresponding residual extensions are

$$\mathbf{e}_b^{0,1} = 10^{-4} \times [0.1569 \quad 0.1569 \quad 0.0116 \quad 0.0116]^T, \quad (57)$$

which is smaller than  $\mathbf{e}_b^{0,0}$ . Repeating the correction step, and it gives

$$\mathbf{e}_b^{0,2} = 10^{-7} \times [0.1051 \quad 0.1711 \quad 0.0122 \quad 0.0005]^T, \quad (58)$$

$$\mathbf{e}_b^{0,3} = 10^{-13} \times [0.5429 \quad 0.8926 \quad 0.0644 \quad 0.0033]^T, \quad (59)$$

and stops until  $\|\mathbf{e}_b^{0,3}\| < \epsilon_c = 10^{-12}$ . Note that all matrices are updated based on the current configuration  $\mathbf{x}^{0,j}$ . Then  $\mathbf{x}^{0,3}$  is regarded as extension-free and assigned as  $\mathbf{x}^2$ . Repeating incrementation and correction steps until

$$\|\mathbf{x}_t - \mathbf{x}^i\| < \epsilon_t = 0.02. \quad (60)$$

It results in a set of compatible configuration points, noted as  $[\mathbf{x}_A^0, \dots, \mathbf{x}_A^{160}]$ , where the subscript A is added to represent Scheme A. This forms a kinematic path which is noted as Path A, and it can be parameterized also in terms of  $[l_1, l_2, l_3]$  (computed by the distance formula) and  $[\phi_1, \phi_2, \phi_3]$  (computed by Eq. (4)) respectively as in Fig. 9, and correspondingly gives  $[\mathbf{l}_A^0, \dots, \mathbf{l}_A^{160}]$  and  $[\phi_A^0, \dots, \phi_A^{160}]$ .

#### 4.1.3. Scheme B: employing three rotational actuators

As the preference is to approach the target configuration fastest in terms of hinge rotations, the method presented in Section 3.3 is employed. The target rotation for the initial step is

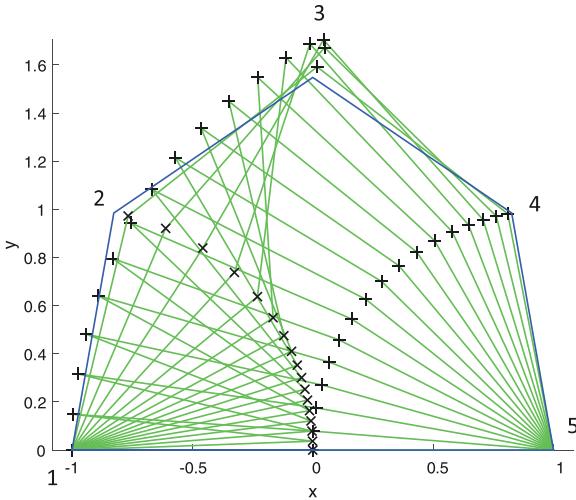
$$\phi_t - \phi^0 = \begin{bmatrix} 3.9397 \\ -1.9452 \\ 0.7981 \end{bmatrix}. \quad (61)$$

The compatibility matrix for rotational hinges based on Eq. (13) is given as

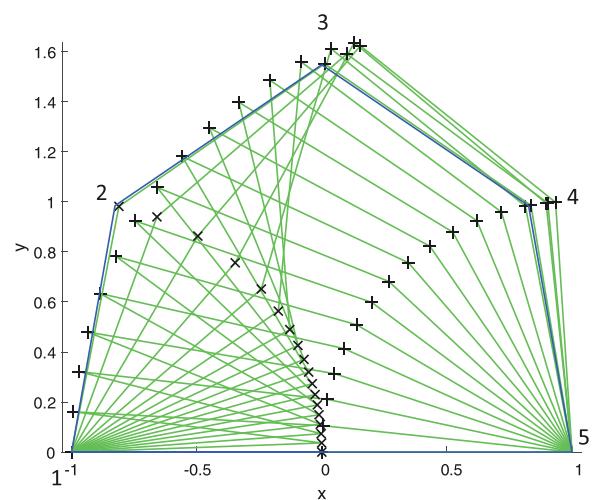
$$\mathbf{C}^h|_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 2 \end{bmatrix}. \quad (62)$$

Based on Eqs. (30) and (31),

$$\hat{\mathbf{d}}_m^0 = \mathbf{W}_{r_h}^h|_0 (\mathbf{V}_{r_h}^h)_{0,0}^{-1} (\mathbf{U}_{r_h}^h)_{0,0}^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.0000 \\ 0.4237 \\ -0.0000 \\ 3.9397 \\ -0.0000 \\ 2.3689 \end{bmatrix} \quad (63)$$



(a) Configurations in Path A generated by Scheme A



(b) Configurations in Path B generated by Scheme B

Fig. 10. Configurations of the 5-bar linkage in the deployment process.

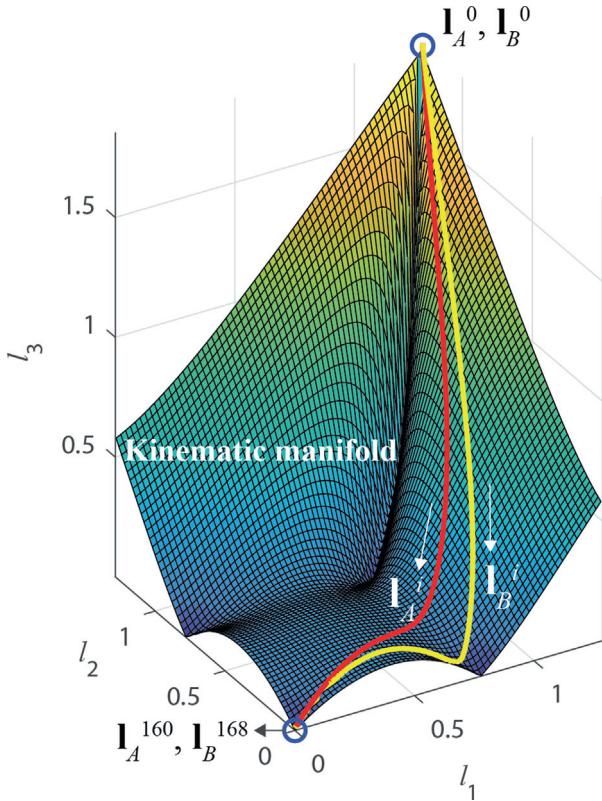


Fig. 11. In the space parameterized by  $[l_1, l_2, l_3]$ , plots of the kinematic manifold, the kinematic path  $I_A^i$  generated by Scheme A, and the kinematic path  $I_B^i$  generated by Scheme B.

and

$$\mathbf{U}^h|_0 \mathbf{V}^h|_0 (\mathbf{W}^h)_0^T = \begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^{rh} \end{bmatrix}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 2 \end{bmatrix}, \quad (64)$$

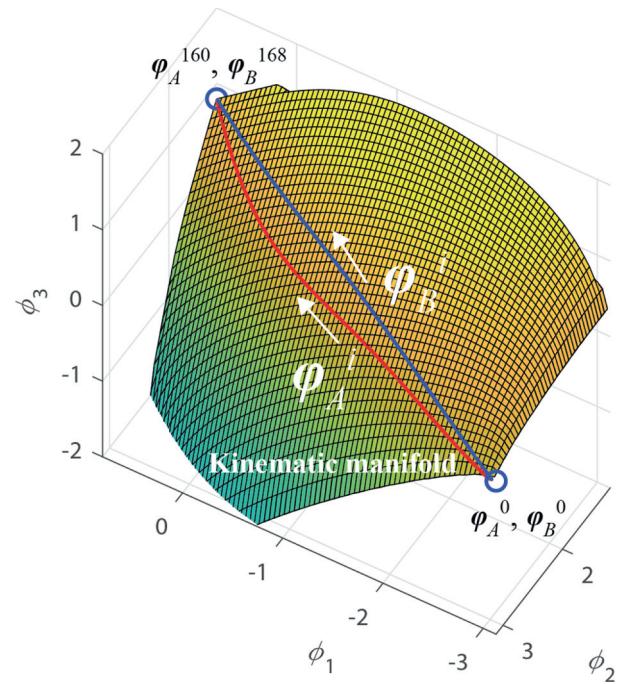
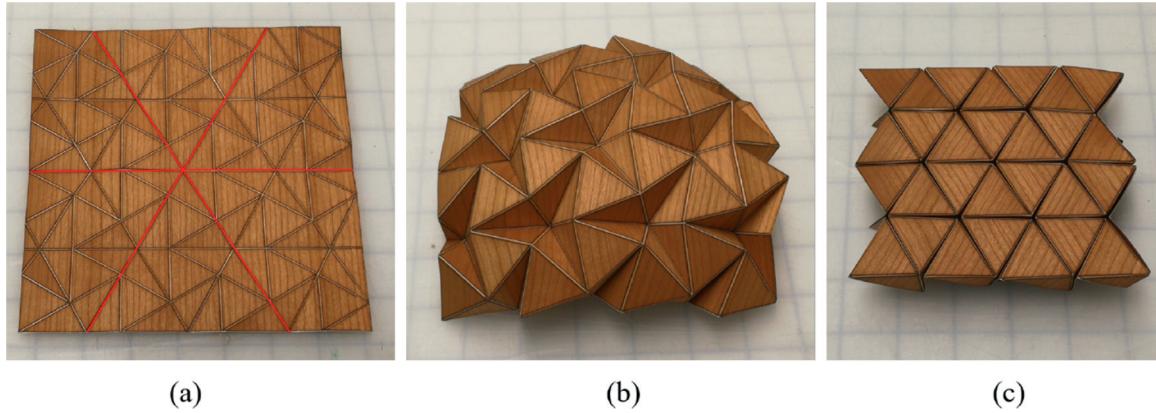


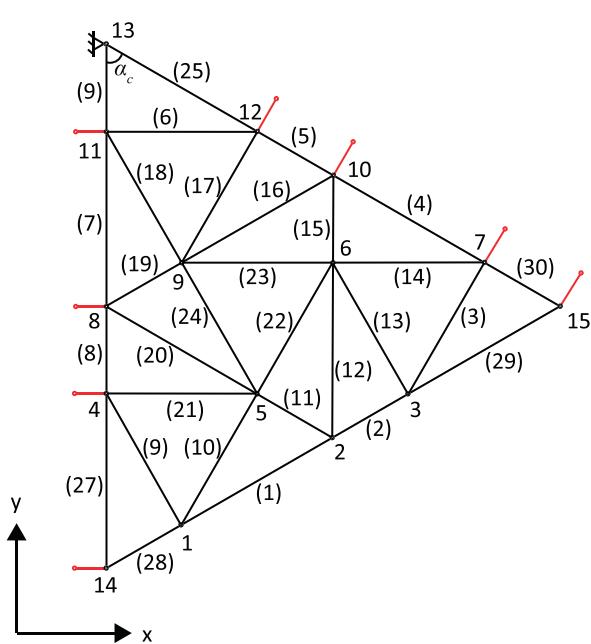
Fig. 12. In the space parameterized by  $[\phi_1, \phi_2, \phi_3]$ , plots of the kinematic manifold, the kinematic path  $\phi_A^i$  generated by Scheme A, and the kinematic path  $\phi_B^i$  generated by Scheme B.

where

$$\mathbf{W}_{\phi_h}^h|_0 = \begin{bmatrix} 0 & -0.5000 & 0.7071 & 0 & -0.5000 & 0 \\ -0.1721 & 0 & 0 & 0.6318 & 0 & -0.7558 \\ -0.0000 & 0.7071 & -0.0000 & -0.0000 & -0.7071 & -0.0000 \\ -0.4271 & -0.0000 & -0.0000 & -0.7392 & 0.0000 & -0.5207 \\ 0.0000 & -0.5000 & -0.7071 & 0.0000 & -0.5000 & -0.0000 \\ 0.8877 & 0.0000 & 0.0000 & -0.2332 & 0.0000 & -0.3971 \end{bmatrix}, \quad (65)$$



**Fig. 13.** Configurations of the triangular Resch's pattern that are (a) unfolded, (b) intermediate, and (c) fully folded.



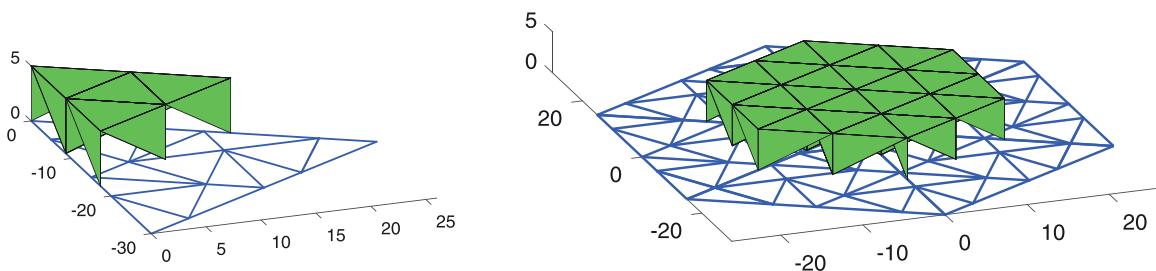
**Fig. 14.** The numbered bar-plate structure of a symmetric unit (noted in Fig. 13(a) by red lines) of the triangle Resch's pattern, where the number without bracket in the figure indicates the number of a node and the number with bracket implies the number of a bar. Note that no non-rotational constraint is imposed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$(\mathbf{V}_{r_h^0}^h)_0^{-1} = \begin{bmatrix} 0.4030 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5412 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7071 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8546 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.3066 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.4516 \end{bmatrix}, \quad (66)$$

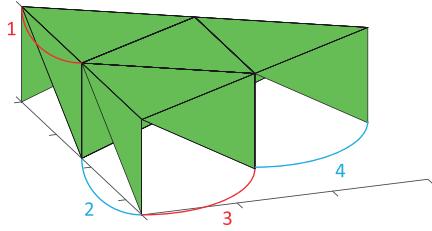
$$(\mathbf{U}_{r_h^0}^h)^T = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.1721 & -0.4271 & 0.8877 \\ -0.2706 & -0.6533 & -0.6533 & 0.2706 & -0.0000 & -0.0000 & 0.0000 \\ 0.5000 & 0.5000 & -0.5000 & 0.5000 & 0.0000 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.6318 & 0.7392 & 0.2332 \\ -0.6533 & 0.2706 & 0.2706 & 0.6533 & 0.0000 & -0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.7558 & -0.5207 & -0.3971 \end{bmatrix}, \quad (67)$$

and  $r_h^0 = 6$ . Normalizing  $\hat{\mathbf{d}}_m^0$  gives  $\mathbf{d}_m^0$ , and moving  $\mathbf{x}^0$  along  $\mathbf{d}_m^0$  by  $\alpha = 0.02$  gives the incrementation as

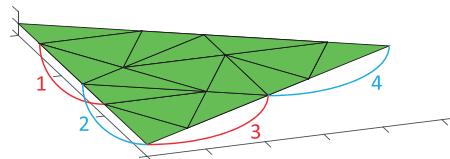
$$\mathbf{x}^{0,0} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + 0.02 \begin{bmatrix} 0 \\ 0 \\ -0.0000 \\ 0.0918 \\ -0.0000 \\ 0.8534 \\ -0.0000 \\ 0.5131 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1.0000 \\ 0 \\ 0 \\ -0.0000 \\ 0.0018 \\ -1.0000 \\ 0.0171 \\ -0.0000 \\ 0.0103 \\ 1.0000 \\ 0 \end{bmatrix}. \quad (68)$$



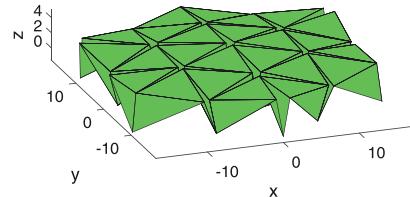
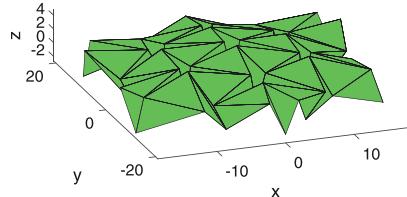
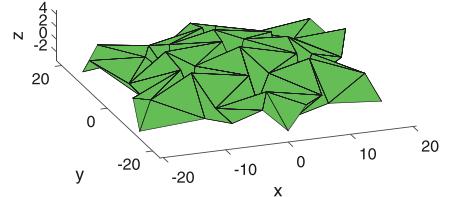
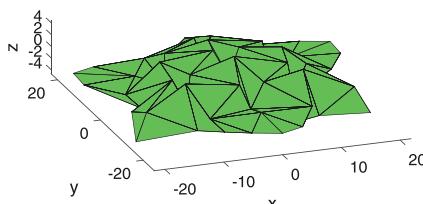
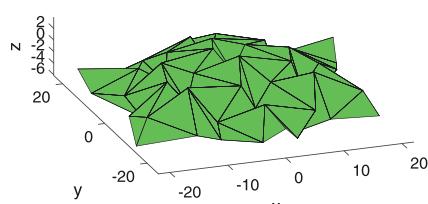
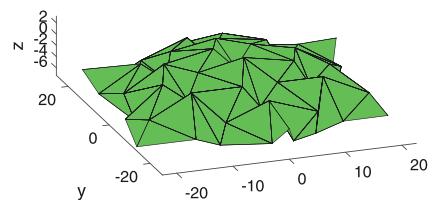
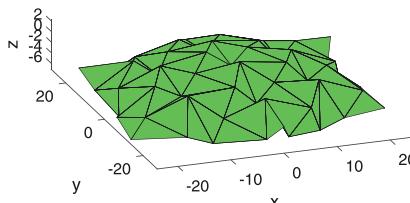
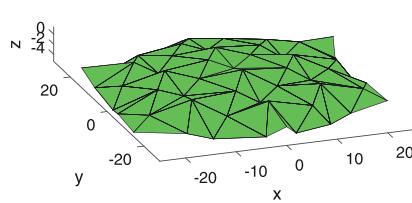
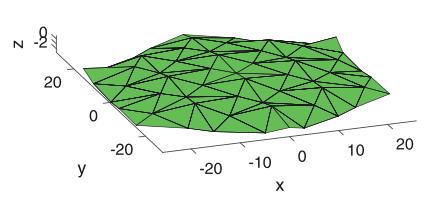
**Fig. 15.** Assemblage based on symmetry where blue lines represent the flat configuration and green plates denote the fully folded configuration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) In folded configuration



(b) In flat configuration

(a)  $\mathbf{x}_c^{12}$ (b)  $\mathbf{x}_c^{24}$ (c)  $\mathbf{x}_c^{36}$ (d)  $\mathbf{x}_c^{48}$ (e)  $\mathbf{x}_c^{60}$ (f)  $\mathbf{x}_c^{72}$ (g)  $\mathbf{x}_c^{84}$ (h)  $\mathbf{x}_c^{96}$ (i)  $\mathbf{x}_c^{108}$ 

**Fig. 17.** Selected configurations on path 1 where (a) is slightly unfolded from the fully folded state and (i) is almost flat. The superscript of  $\mathbf{x}_c^i$  indicates the number increment is  $i$ .

Iterative correction and incrementation steps can be applied as in Section 4.1.2. A kinematic path, noted as *Path B*, is generated, and it can be represented as  $[\mathbf{x}_B^0, \dots, \mathbf{x}_B^{168}], [\mathbf{l}_B^0, \dots, \mathbf{l}_B^{168}]$ , and  $[\phi_B^0, \dots, \phi_B^{168}]$ , where subscript *B* is to represent Scheme B.

#### 4.1.4. Comparison of paths A and B

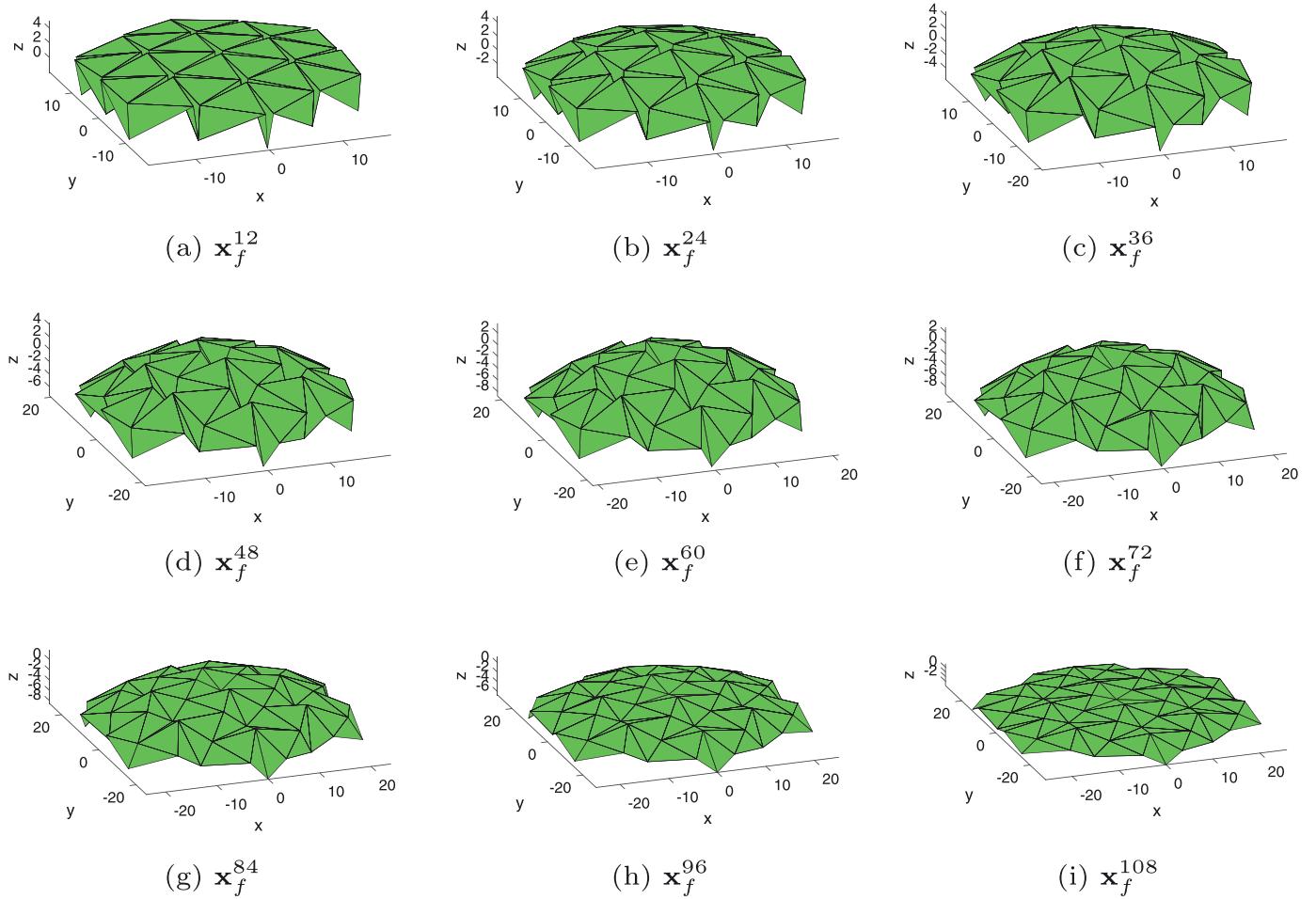
Configurations in the two paths are given in Fig. 10. Differences can be observed, for instance, the variation of the fold angle of hinge 5 is monotonic in Path A, but non-monotonic in Path B. In Fig. 11, the space is parameterized by  $[l_1, l_2, l_3]$ , and Path A and B represented by  $\mathbf{l}_A^i$  and  $\mathbf{l}_B^i$  are plotted. The kinematic manifold is generated by assigning different values to  $[l_1, l_2]$  and then compute corresponding compatible values of  $l_3$  according to the analytical kinematic relationship which is only available for simple mechanisms. Many other bifurcation manifolds are not

plotted. Similarly, Path A ( $\phi_A^i$ ), Path B ( $\phi_B^i$ ), and the kinematic manifold are plotted in the space parameterized by  $[\phi_1, \phi_2, \phi_3]$  in Fig. 12. Visually, both paths are quite short, while Path A is shorter in the space of  $[l_1, l_2, l_3]$  and slightly longer in the space of  $[\phi_1, \phi_2, \phi_3]$ , which can be verified by computing the lengths of paths. Lengths of the two paths in  $[l_1, l_2, l_3]$  respectively are

$$\sum_{i=1}^{160} \| \mathbf{l}_A^i - \mathbf{l}_A^{i-1} \| = 2.8740, \quad (69)$$

and

$$\sum_{i=1}^{168} \| \mathbf{l}_B^i - \mathbf{l}_B^{i-1} \| = 3.0993. \quad (70)$$



**Fig. 18.** Selected configurations on path 2 where (a) is slightly unfolded from the fully folded state and (i) is almost flat. The superscript of  $\mathbf{x}_f^i$  indicates the number increment is  $i$ .

Lengths of the two paths in  $[\phi_1, \phi_2, \phi_3]$  respectively are

$$\sum_{i=1}^{160} \|\phi_A^i - \phi_A^{i-1}\| = 4.5246, \quad (71)$$

and

$$\sum_{i=1}^{168} \|\phi_B^i - \phi_B^{i-1}\| = 4.4909. \quad (72)$$

Fig. 6 is the schematic version of Figs. 11 and 12, and all of them present the essential idea that is to walk within the manifold and try to approach the target quickly. To conclude the plots:

- different kinematic paths can be found using the shooting method employing different target functions based on user preferences;
- a relatively shorter path (not necessarily the shortest) is found in the specific parametric space in which the shooting target is assigned, while it appears to be relatively longer in other parametric spaces;
- the kinematic manifold of a mechanism that employs revolute joints has a simpler/smooth structure in the space parameterized by angles than in the space parameterized by coordinates or distances.

Implications for practical actuators from the result of this analysis are:

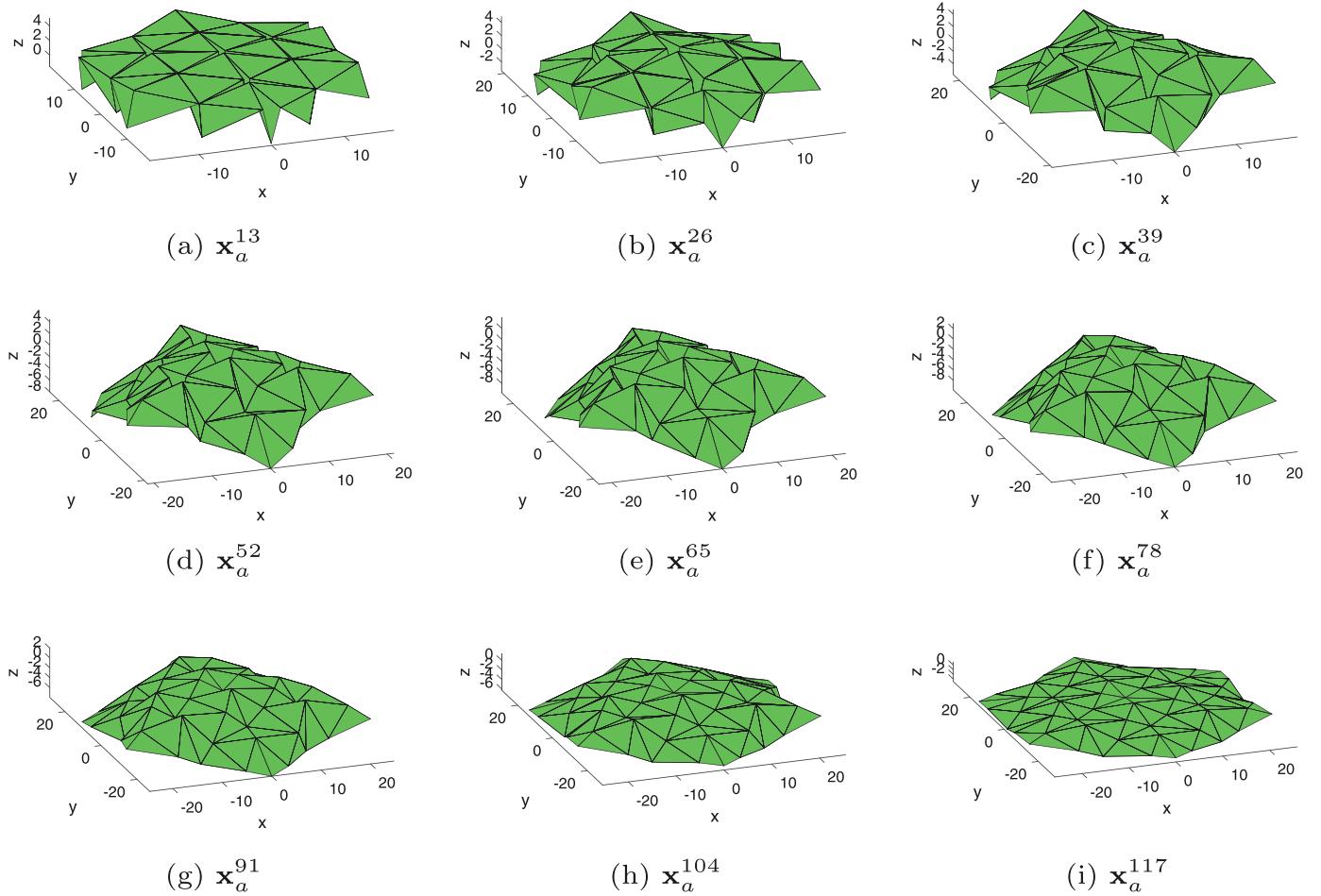
- if three displacement-controlled linear actuators are employed as in Fig. 9(a), controlling the actuators' lengths ( $[l_1, l_2, l_3]$ ) according to path  $\mathbf{l}_A^i$  in Fig. 11 gives a compatible and faster actuation process (than  $\mathbf{l}_B^i$ );

- if three angle-controlled rotational actuators are used as in Fig. 9(b), controlling the actuators' angles ( $[\phi_1, \phi_2, \phi_3]$ ) according to path  $\phi_B^i$  in Fig. 12 gives a compatible and faster actuation process (than  $\phi_A^i$ ).

#### 4.2. Folding paths of a multi-DOF origami mechanism

To demonstrate the generality of the method, a more complex example, for which the analytical kinematic relationship is not available, is demonstrated in this subsection. The triangular Ron Resch's pattern has multi-DOF behavior. It can be folded from a completely flat geometry into a spherical shape and then further into a fully folded flat configuration, as shown in Fig. 13. This section presents a procedure for finding three different paths that are represented by many intermediate configuration points between the complete-flat configuration (Fig. 13(a)) and the fully folded configuration (Fig. 13(c)).

Due to the existence of complex bifurcation paths in the complete-flat configuration, it is easier to set it as the target configuration  $\mathbf{x}_t$  and to set the fully folded state as the initial configuration  $\mathbf{x}^0$ . This choice can avoid the mechanism to diverge into other branches of the kinematic manifold. Bifurcation paths can be usually avoided by (1) changing the target function, (2) changing the parametric space (of the target function), (3) swapping the path-finding direction (set the bifurcation point as the target configuration), and (4) introducing contacts to block unwanted kinematic branches. As long as a path is found, it can be used in both directions for actuation and so on. Introducing contact blocks and



**Fig. 19.** Selected configurations on path 3 where (a) is slightly unfolded from the fully folded state and (i) is almost flat. The superscript of  $\mathbf{x}_a^i$  indicates the number increment is  $i$ .

initial imperfections/perturbations are common ways to avoid bifurcation in the practical actuation.

The plane symmetry can be identified from this pattern (as noted with the red lines in Fig. 13(a)). This symmetry can be preserved in the folding motion if it is folded symmetrically. The first step is to transform its crease pattern into a numbered bar-plate structure. This can be done automatically using Tesslatica [37]. A symmetrical unit is taken out and shown in Fig. 14.

#### 4.2.1. Input

The nodal coordinates of the initial configuration, which is depicted in green in Fig. 15, and the target configuration, which is denoted with the blue lines in Fig. 15, are  $\mathbf{x}^0$  and  $\mathbf{x}_t$ , respectively, and can be quickly calculated based on their geometries. Complete configurations can be assembled as in Fig. 15b. In Fig. 14, node 13 is pinned to the foundation, and its coordinates are (0,0,0). The length of bar 1 (connecting nodes 1 and 2) is 10, and  $\alpha_c$  equals 60°. The reaction bars at the two sides are used to mimic the symmetric/sliding boundary condition. There are 15 nodes, 30 pin-jointed bars, and 8 boundary nodes (11, 8, 4, 14, 12, 10, 7, 15). Contact constraints are considered in Based on Selected Target Nodal Distances in Section 4.2.2. The simulation parameters are set as follows,  $\epsilon_c = 10^{-12}$ ,  $\epsilon_t = 0.5$ , and  $\alpha = 0.5$ . The numerical values of most of the matrices and vectors used in this section are provided in Appendix A.7.

#### 4.2.2. Simulation

The total DOF (before considering any constraints) is  $3n = 3 \times 15 = 45$ . Due to the supports at nodes 13 (x, y, and z directions), 11 (x di-

rection), 8 (x direction), 4 (x direction), and 14 (x direction), there are 7 DOF that can be removed directly, and the remaining total DOF is 38. Three different ways of finding a path between  $\mathbf{x}^0$  and  $\mathbf{x}_t$  that are illustrated in Sections 3.1.1 and 3.3, are demonstrated below.

##### Based on target nodal coordinates

Target displacements, which is following Section 3.1.1, in the first step are given as

$$\mathbf{d}_t^0 = \mathbf{x}_t - \mathbf{x}^0. \quad (73)$$

Based on  $\mathbf{x}^0$ ,  $\mathbf{C}^b|_0$  with size  $30 \times 38$  and  $\mathbf{C}^c|_0$  (for nodes 12, 10, 7, and 15) with size  $4 \times 38$  can be computed using Eqs. (2) and (15). The SVD of  $\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^c \end{bmatrix}|_0$  gives  $\mathbf{W}_m|_0$ , which has size  $38 \times 4$  and indicates that there are four kinematic DOF, based on Eqs. (25) and (26). As shown in Eq. (27), projecting  $\mathbf{d}_t^0$  into  $\mathbf{W}_m|_0$  gives

$$\hat{\mathbf{d}}_m^0 = \mathbf{W}_m|_0 (\mathbf{W}_m|_0)^T \mathbf{d}_t^0. \quad (74)$$

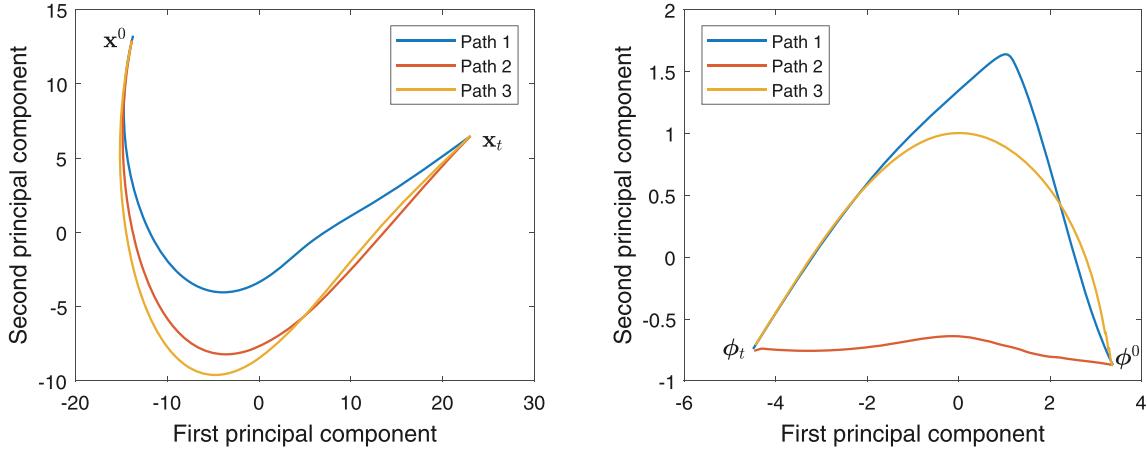
and  $\hat{\mathbf{d}}_m^0$  is achieved by normalizing  $\hat{\mathbf{d}}_m^0$ . Moving  $\mathbf{x}_0$  along  $\hat{\mathbf{d}}_m^0$  by  $\alpha$  gives

$$\mathbf{x}^{0,0} = \mathbf{x}^0 + \alpha \hat{\mathbf{d}}_m^0. \quad (75)$$

Extensions of all bars  $\mathbf{e}_b^{0,0}$  and distances of all boundary nodes to the sliding plane  $\mathbf{e}_c^{0,0}$  can be computed based on the coordinates  $\mathbf{x}^{0,0}$ , which are higher-order residuals and

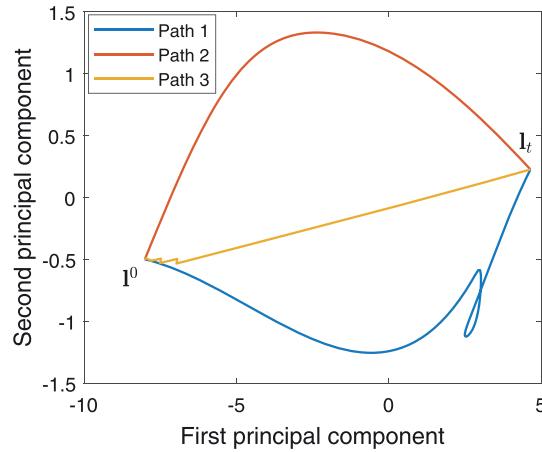
$$\left\| \begin{bmatrix} \mathbf{e}_b^{0,0T} & \mathbf{e}_c^{0,0T} \end{bmatrix} \right\| = 4.7 \times 10^{-3}. \quad (76)$$

Updating the compatibility matrices  $\mathbf{C}^b|_{0,0}$  and  $\mathbf{C}^c|_{0,0}$  and the corresponding SVD gives the following correction displacement based on



(a) In coordinate configuration space  $\mathbf{x}^i$ , where each configuration is parameterized by coordinates of all the nodes and correspondingly represented by a high-dimensional point consisting of those nodal coordinates

(b) In fold-angle configuration space  $\phi^i$ , where each configuration is parameterized by fold angles of all the hinges and correspondingly represented by a high-dimensional point consisting of the values of those fold angles



(c) In “actuator” configuration space  $\mathbf{l}^i$ , where each configuration is parameterized by the lengths of “actuators” (distances between selected nodes) and correspondingly represented by a high-dimensional point consisting of those lengths of “actuators”

**Fig. 20.** Paths 1, 2, and 3 plotted on the first two principal components in (a) coordinate, (b) fold-angle, and (c) “actuator” configuration space, respectively.

Eq. (35)

$$\mathbf{d}_c^{0,0} = -\mathbf{W}_{r_C^{0,0}}|_{0,0}(\mathbf{V}_{r_C^{0,0}})^{-1}|_{0,0}(\mathbf{U}_{r_C^{0,0}})|_{0,0} \begin{bmatrix} \mathbf{e}_b \\ \mathbf{e}_c \end{bmatrix}_{0,0}, \quad (77)$$

where  $r_C^{0,0} = 34$ . The configuration point with one correction iteration is

$$\mathbf{x}^{0,1} = \mathbf{x}^{0,0} + \mathbf{d}_c^{0,0}. \quad (78)$$

Evaluating the norm of constraint violation  $[\mathbf{e}_b^{0,1T} \mathbf{e}_c^{0,1T}]$  of  $\mathbf{x}^{0,1}$  gives  $3.1 \times 10^{-7}$ . Repeating the correction step until  $j = 2$  gives

$$\left\| \begin{bmatrix} \mathbf{e}_b^{0,2T} & \mathbf{e}_c^{0,2T} \end{bmatrix} \right\| = 9.6 \times 10^{-15} < \epsilon_c. \quad (79)$$

As a result,

$$\mathbf{x}^1 = \mathbf{x}^{0,2}. \quad (80)$$

Now, repeating this process until  $i = 119$  results in

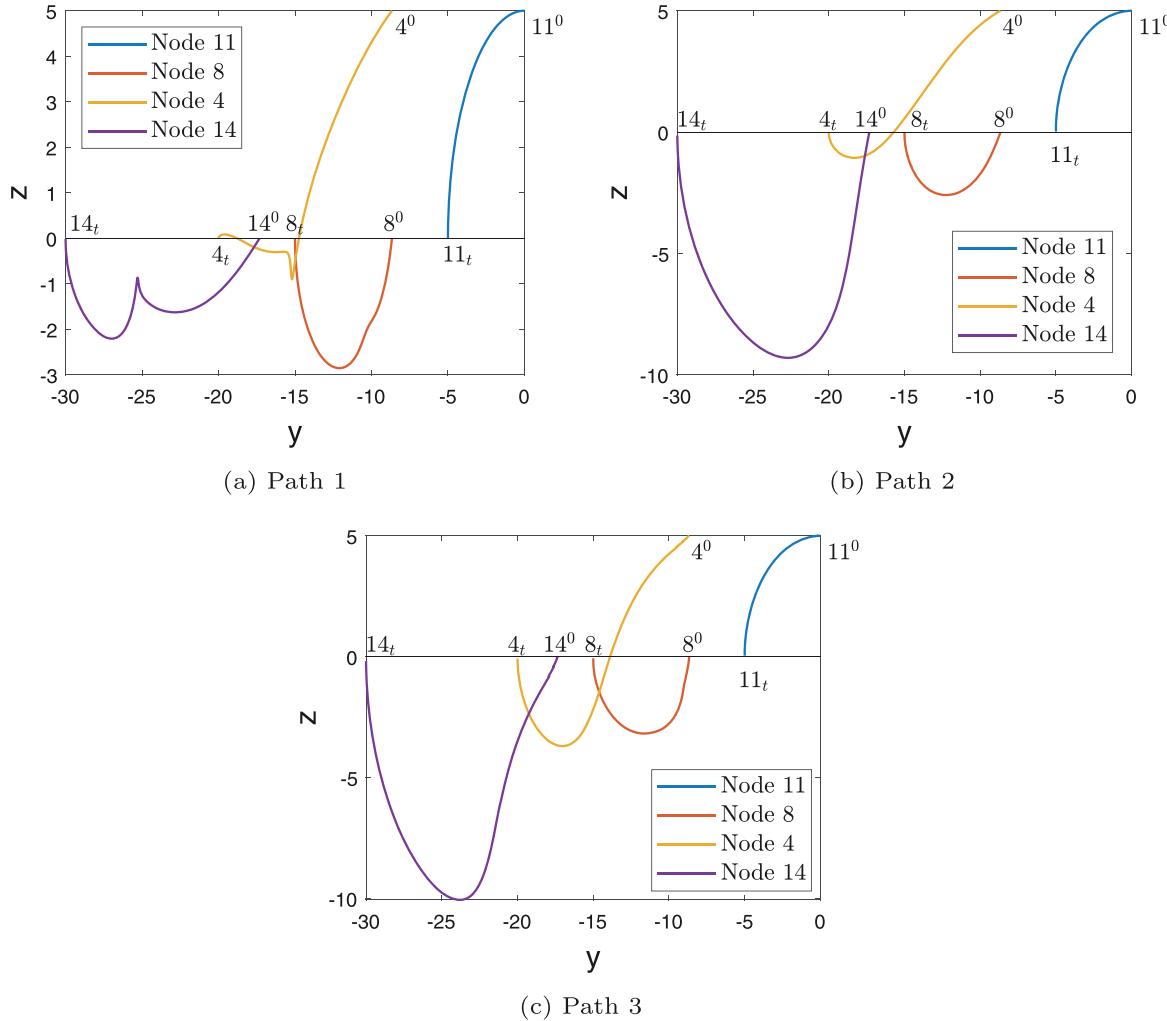
$$\|\mathbf{x}_t - \mathbf{x}^{119}\| = 0.06 < \epsilon_t \quad (81)$$

and  $[\mathbf{x}_c^1, \dots, \mathbf{x}_c^{119}]$ , where the subscript  $c$  is added to differentiate it from other paths, gives one path that connects  $\mathbf{x}^0$  and  $\mathbf{x}_t$ .  $[\phi_c^1, \dots, \phi_c^{119}]$  can be correspondingly computed using Eq. (4), and it describes the same path parameterized by the fold angles. The process presented in this subsection is identical to that in the flowchart shown in Fig. 7, and contact constraint is not considered.

*Based on target fold angles*

According to Section 3.3,  $\mathbf{C}^h|_0$ , which has the size of  $18 \times 38$ , needs to be computed (additional to other matrices in Section 4.2.2) based on Eq. (8). Here,  $\phi^0$  and  $\phi_t$  can be computed using Eq. (4). The SVD of  $\begin{bmatrix} \mathbf{C}^b \\ \mathbf{C}^c \\ \mathbf{C}^h \end{bmatrix}|_0$  (as shown in Eq. (31)) gives

$$\hat{\mathbf{d}}_m^0 = \mathbf{W}_{r_h^0}|_0(\mathbf{V}_{r_h^0})^{-1}|_0(\mathbf{U}_{r_h^0})^T \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \phi_t - \phi^0 \end{bmatrix}, \quad (82)$$



**Fig. 21.** Trajectories of nodes 11, 8, 4, and 14 in  $yz$  plane during the (un)folding process where the 0 superscript and  $t$  subscript in the plots indicate the nodal positions of the initial and target configurations, respectively.

using Eq. (30) where  $r_h^0 = 38$ . The remaining steps are the same as those in Section 4.2.2.

As a result, a path consisting of  $[x_f^1, \dots, x_f^{122}]$  (or  $[\phi_f^1, \dots, \phi_f^{122}]$ ) is achieved where the subscript  $f$  denotes the path found based on the target fold angles, and contact constraint is not considered.

#### Based on selected target nodal distances

This subsection follows the same principle given in Section 3.3. Place four linear “actuators” between the following nodes

$$\begin{bmatrix} 4 & 11 \\ 8 & 14 \\ 2 & 14 \\ 2 & 15 \end{bmatrix} \quad (83)$$

with initial and final lengths of

$$l^0 = \begin{bmatrix} 8.67 \\ 8.67 \\ 8.67 \\ 8.67 \end{bmatrix}, \quad l_t = \begin{bmatrix} 15 \\ 15 \\ 15 \\ 15 \end{bmatrix}, \quad (84)$$

as illustrated in Fig. 16 where curves are used to represent linear “actuators” for better clarity.

Compatibility matrix of linear “actuators” at  $x^0$ , noted as  $C^a|_0$  that has size  $4 \times 38$ , can be computed based on Eq. (2) and (83). Compati-

bility constraints are

$$\begin{bmatrix} C^b \\ C^c \\ C^a \end{bmatrix} \hat{d}_m^0 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ l_t - l^0 \end{bmatrix}. \quad (85)$$

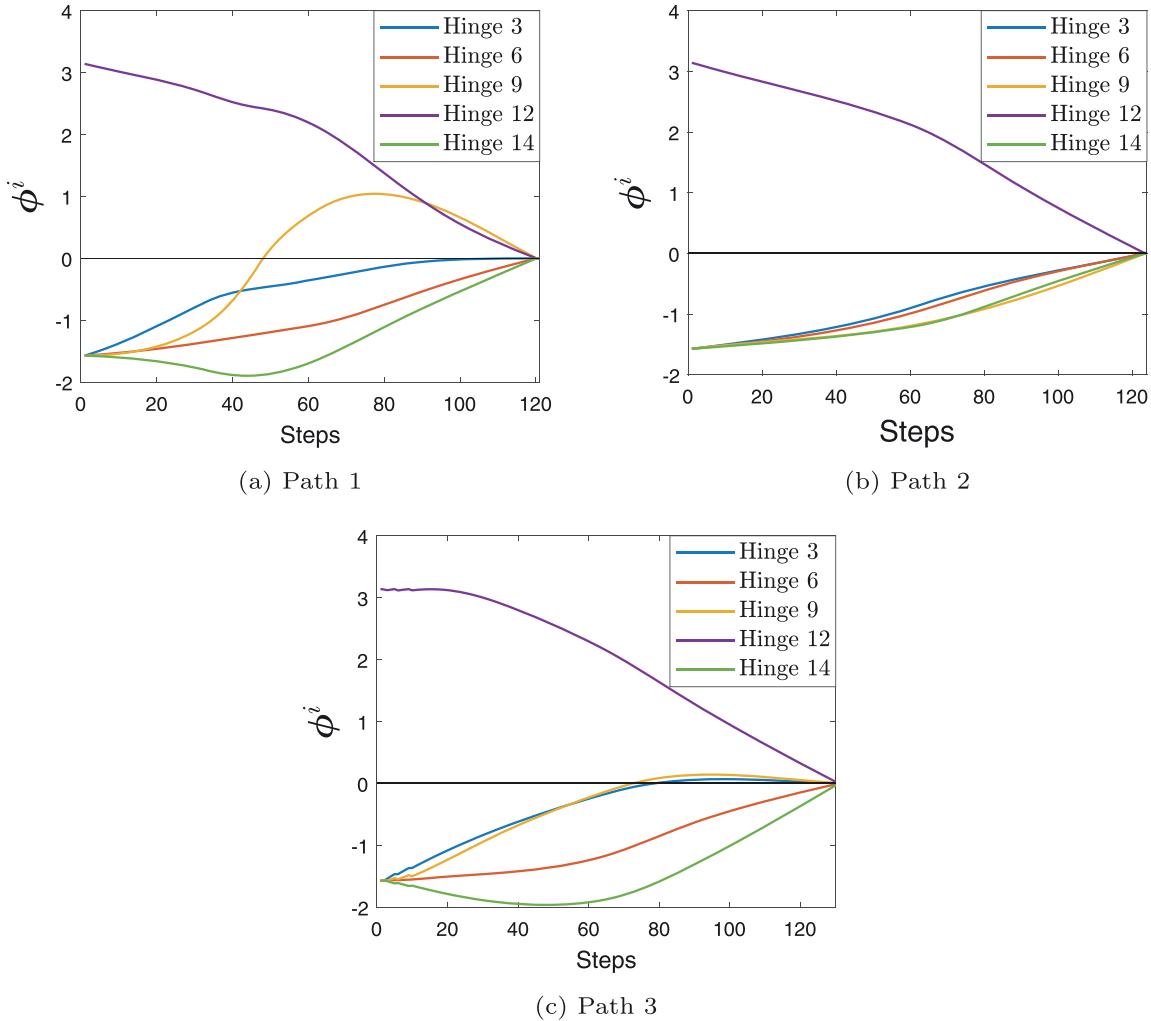
Contact constraints are considered in this path-finding process by employing Eq. (28) so that

$$C_{ch}|_0 \hat{d}_m^0 < 0. \quad (86)$$

By minimizing  $\|\hat{d}_m^0\|$  subject to Eqs. (85) and (86), which is done with function `quadprog` in Matlab, a unique solution can be found. Correction displacements are generated using Eqs. (34) and (35). As a result, a path is found and can be represented by  $[x_a^1, \dots, x_a^{129}]$  in terms of coordinates,  $[\phi_a^1, \dots, \phi_a^{129}]$  in terms of fold angles, and  $[l_a^1, \dots, l_a^{129}]$  in terms of “actuator” lengths, where the subscript  $a$  denotes the path found based on target lengths of “actuators”.

#### 4.2.3. Comparison of the three paths

Denote path 1 that can be represented by  $[x_c^1, \dots, x_c^{119}]$  in coordinate space,  $[\phi_c^1, \dots, \phi_c^{119}]$  in fold-angle space, and  $[l_c^1, \dots, l_c^{119}]$  in “actuator” length space. Similarly, path 2 is defined by  $[x_f^1, \dots, x_f^{122}], [\phi_f^1, \dots, \phi_f^{122}]$ , and  $[l_f^1, \dots, l_f^{122}]$ , and path 3 is defined by  $[x_a^1, \dots, x_a^{129}], [\phi_a^1, \dots, \phi_a^{129}]$ , and  $[l_a^1, \dots, l_a^{129}]$ . Based on symmetry, the original configuration in Fig. 15 can



**Fig. 22.** Selected fold angle (hinge 3, 6, 9, 12, and 14) variations for 3 paths during the (un)folding process where all the fold angles start from either  $\pi$  or  $-\frac{\pi}{2}$  (the fully folded configuration) and end in 0 (the flat configuration).

be assembled as in Fig. 15b. Some selected configurations (after being assembled) on path 1, path 2, and path 3 are shown in Figs. 17–19, respectively. The motions along three paths are observed to be different, and there is no contact penetration. The configuration shown in Fig. 19a has some panels that are in contact with one another at which contact constraints are activated.

Plots of the three paths on the first two principal components [38] in coordinate configuration space are shown in Fig. 20a, where path 1 appears to be the shortest. Plots of the three paths in fold-angle configuration space are given in Fig. 20b, which show that path 2 is significantly shorter and almost linear. Fig. 20c gives paths in “actuator” configuration space, and path 3 is the shortest and most linear.

Specifically, plots of the trajectories of nodes 11, 8, 4, and 14 of the three paths in the  $yz$  plane are given in Fig. 21, which show that the motions of the nodes in path 2 and 3 are smoother. The 0 superscript and  $t$  subscript in the plots indicate the nodal positions of the initial and target configurations, respectively.

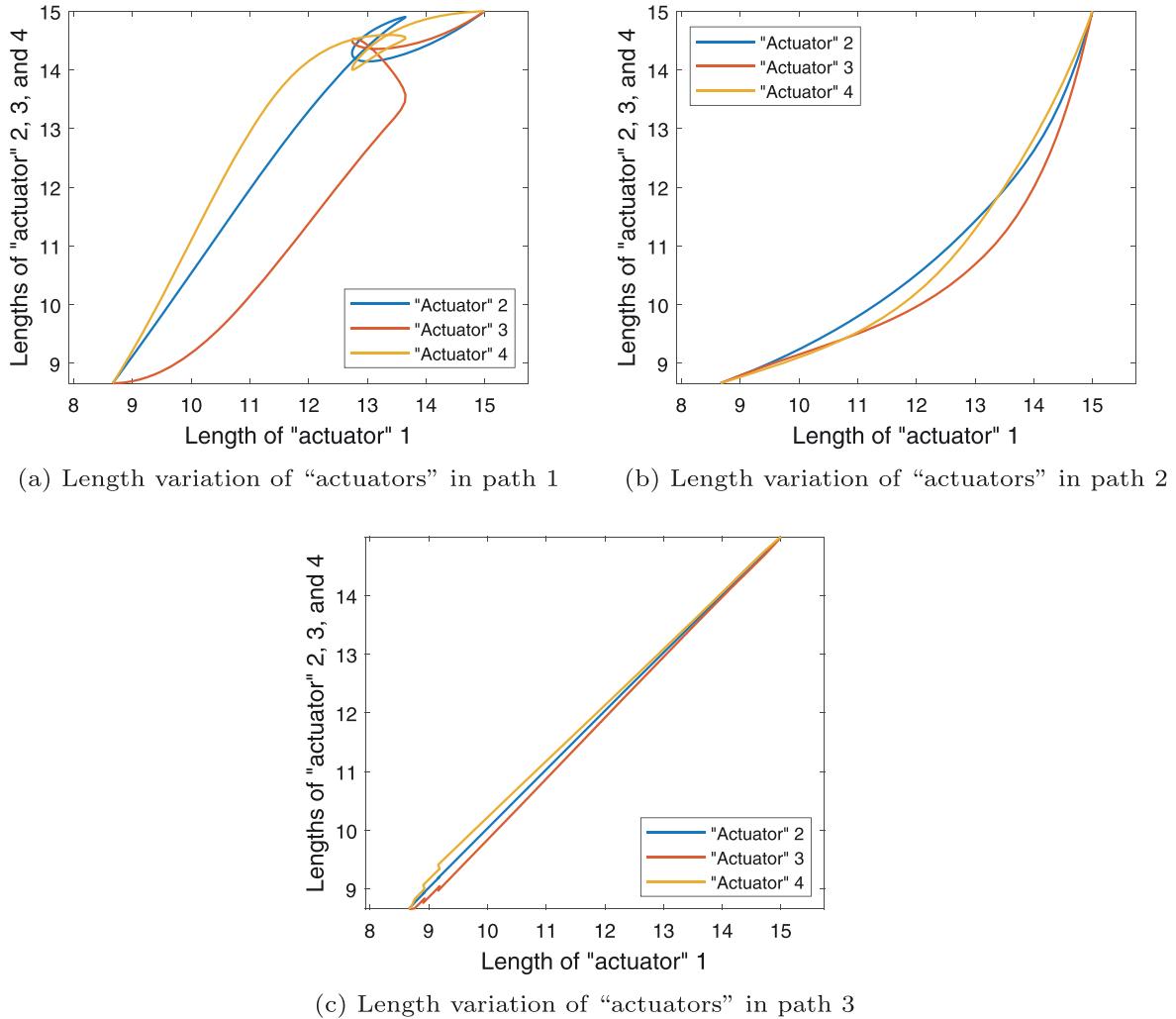
Variations in the fold angles of hinges 3, 6, 9, 12, and 14 for the three paths are plotted in Fig. 22, showing that the angle variations in path 2 are monotonic, while, in path 1 and 2, some are not. Contact constraints are active in path 3. This can be observed by noticing the slight fluctuation in the beginning of the rotation of hinge 12 as shown in Fig. 22c where it attempts to increase beyond  $\pi$  but is constrained by

the contact and pushed back. Variations in lengths of four “actuators” are given in Fig. 23 where path 3 gives the most linear relationship.

Intuitively, using nodal coordinates as the target in this case is similar to pulling the folded geometry apart at each node, thus providing a short path in the coordinate configuration space and involving the flipping of some of the hinges. Using target fold angles here means trying to open all hinges simultaneously, thus generating a smooth (in terms of the nodal trajectory), monotonic (in terms of the fold angle variations), and slightly longer path in coordinate configuration space while producing an almost linear path in fold-angle configuration space. Similarly, setting lengths of four “actuators” as the target tries to vary the corresponding nodal distances uniformly while satisfying kinematic and contact constraints.

Proper assignments of constraints and target direction/deformation functions can lead to desired paths that suit specific applications. For instance, consider installing 4 linear actuators as shown in Fig. 16, paths 1, 2, and 3 can be achieved when controlling their lengths according to Fig. 23a–c respectively. Apparently, employing path 3 gives the easiest actuation control and shortest execution time. If angular actuators are mounted in all hinges, it is easier to follow path 2 where only monotonic motion and shorter variation range are needed as illustrated in Fig. 22.

Movies of simulations and Matlab codes are provided in supplementary materials.



**Fig. 23.** “Actuator” length variations for 3 paths during the (un)folding process where all lengths start from 8.67 and end in 15 as indicated in Eq. (84).

## 5. Conclusion

This paper has presented a path-finding simulation setup that uses a truss model and the shooting method for modeling the motion of multi-DOF mechanisms in a guided way. A target direction/deformation is introduced to help determine a unique motion direction at each increment, providing flexibility in guiding the motion in a desired way by defining the target direction/deformation according to various customized functions. More special constraints can be introduced accordingly, such as rotational relationships between hinges and different boundary conditions. As a consequence, the kinematic manifold can be explored in a more arbitrary way than made possible by existent techniques. This contributes to a better knowledge of valid kinematic paths and is useful in many ways. For instance, the acquisition of a path between two configurations informs how to actuate it from one configuration to another via displacement control (or force control if based on Section 3.1.2). Different actuators suit different paths. For example, it is better to employ linear actuators to follow path 1 and 3 in Section 4.2.3 while it is more suitable to follow path 2 when angular actuators are used.

A major limitation of the shooting method used in this paper is its likelihood of getting stuck in a local minimum. This can be caused by (1) a highly nonlinear and non-convex kinematic manifold that has stationary points where their tangent space is orthogonal to the target direction and (2) contact constraints together with the kinematic manifold that form convex shapes with regard to the target direction at some positions of the configuration space. A solution provided in this paper (same

as avoiding bifurcation paths) is to try to (1) change the target function, (2) change the parametric space (of the target function), (3) swap the path-finding direction, and (4) introduce additional constraints to block unwanted kinematic branches. Another possible solution, which is not covered in this paper, is to sample many points in the manifold and rearrange them in a smooth manner by using (modified versions) of the nudged elastic band method [39]. Then the shooting method can be employed to form a path by setting those sampling points as sequential targets. There are other interesting and important kinematic problems that will be addressed in future works, such as bifurcation in a multi-DOF manifold, design of mechanisms numerically with specified motion paths and DOF, and so on.

There is a disadvantage to multi-DOF mechanisms in practice is that they are “floppy” and require many actuators to control. This disadvantage can be resolved by introducing elastic components into the system to guide the actuation. The proper design of elastic members can generate multi-stability in the incorporated structure that can be reconfigured with significantly fewer actuators than the structure’s original kinematic DOF. This work is presented in Li and Pellegrino [23].

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Discussions with Professor Sergio Pellegrino, Professor Kaushik Bhattacharya, Dr Robert Lang, Hao Zhou, Charles Dorn, and Dr Fufu Yang are acknowledged. This paper is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-18-1-0566 directed by Dr Ken Goretta.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.ijmecsci.2020.105709](https://doi.org/10.1016/j.ijmecsci.2020.105709).

## Appendix A

### A1. Compatibility matrix for pin-jointed bars

```

function EQUIL=eqtr3_m(NODE,ELEM)
% The function EQTR3(NODE,ELEM) assembles the
% compatibility matrix of any 3D truss.
% Form equilibrium matrix and then transpose it to
% compatibility matrix
% NODE (dimension nnodes*6) stores the X,Y,Z coordinates
% of each node in col. 1,2,3, and the constraints in X,Y
% ,Z direction in col. 4,5,6 (0=free, 1=constr.)
%
% ELEM: pin-jointed bar connections
[nnode,var]=size(NODE);
if (var ~= 6)
    error('ERROR in EQTR3: NODE matrix of incorrect size')
end
[nelem,~]=size(ELEM);
% Initialize matrix ROWNO (dimension nnodes*3)
% ROWNO(i,j)=0 => displ. of node i in dir. j is
% constrained, hence no contribution to EQUIL
% ROWNO(i,j)=n => contribution of d.o.f. j of node i to
% be stored in row no. n of EQUIL
icount=0;
for i=1:nnode
    for j=1:3
        if (NODE(i,j+3) == 0)
            icount=icount+1;
            ROWNO(i,j)=icount;
        else
            ROWNO(i,j)=0;
        end
    end
end
ndof=icount;
EQUIL=zeros(ndof,nelem);
icol=0;
for ielem=1:nelem
    icol=icol+1;
    C=[NODE(ELEM(ielem,1),1:3),NODE(ELEM(ielem,2)
        ,1:3)];
    len=sqrt((C(1)-C(4))^2+(C(2)-C(5))^2+(C(3)-C(6))^2);
    x=(C(1)-C(4))/len;
    y=(C(2)-C(5))/len;
    z=(C(3)-C(6))/len;
    EQ1=[x y z -x -y -z];
    ii=0;
    for i=1:2
        for j=1:3
            ii=ii+1;
            irow=ROWNO(ELEM(ielem,i),j);
            if (irow > 0)
                EQUIL(irow,icol)=EQ1(ii);
            end
        end
    end
end
EQUIL=EQUIL';%transpose it to compatibility matrix
return

```

### A2. Computation of the fold angle for one hinge

```

function [phi] = config_fold_angle(P1,P2,P3,P4)
% P1=x_i ,P2=x_j ,P3=x_k ,P4=x_l ,P1P2 is the hinge
%mountain -> negative , valley -> positive
n1=cross((P3-P1),(P2-P1))/(norm(P3-P1)*norm(P2-P1));
n1=n1/norm(n1);
n2=cross((P2-P1),(P4-P1))/(norm(P2-P1)*norm(P4-P1));
n2=n2/norm(n2);
m1=cross((P2-P1),(n1))/(norm(P2-P1)*norm(n1));
m1=m1/norm(m1);
x=dot(n1,n2);
y=dot(m1,n2);
phi_temp=atan(abs(y/x));
if x>=0
    if y>=0
        phi=phi_temp;%obtuse valley
    else
        phi=-phi_temp;%obtuse mountain
    end
else
    if y>=0
        phi=pi-phi_temp;%acute valley
    else
        phi=-(pi-phi_temp);%acute mountain
    end
end

```

### A3. Compatibility matrix for one hinge

```

function [c_a] = displacement_rotation(P1_number,
    P2_number, P3_number, P4_number, Points, Points_b)
P1=Points(P1_number,:);% P1=x_i
P2=Points(P2_number,:);% P2=x_j
P3=Points(P3_number,:);% P3=x_k
P4=Points(P4_number,:);% P4=x_l
n1=cross((P3-P1),(P2-P1))/(norm(P3-P1)*norm(P2-P1));
n1u=n1/norm(n1);
n2=cross((P2-P1),(P4-P1))/(norm(P2-P1)*norm(P4-P1));
n2u=n2/norm(n2);
% a is the displacement of P1, b for P2, c for P3, dd for
% P4
syms a1 a2 a3 b1 b2 b3 c1 c2 c3 dd1 dd2 dd3
a=[a1,a2,a3];
b=[b1,b2,b3];
c=[c1,c2,c3];
d=[dd1,dd2,dd3];
L=norm(P2-P1);
d1=norm(n1)*norm(P3-P1);
l1=sqrt(norm(P3-P1)^2-d1^2);
l1=sqrt(norm(P3-P2)^2-d1^2);
d2=norm(n2)*norm(P4-P1);
l2=sqrt(norm(P4-P1)^2-d2^2);
l3=sqrt(norm(P4-P2)^2-d2^2);
P3P4toAxis=(c*n1u')/d1+(d*n2u')/d2;% P3 and P4
P1toP3P4=-(a*n1u')*l1/(d1*L)-(a*n2u')*l3/(d2*L);
P2toP3P4=-(b*n1u')*l2/(d1*L)-(b*n2u')*l4/(d2*L);
thita=P3P4toAxis+P1toP3P4+P2toP3P4;
all_variables=[a1 a2 a3 b1 b2 b3 c1 c2 c3 dd1 dd2 dd3];
C=[];
for i=1:12
    coeff_temp=coeffs(thita, all_variables(i));
    [~,x]=size(coeff_temp);
    if x==1
        coeff{i}=0;
    else
        coeff{i}=coeff_temp(2);
    end
    C=cat(2,C,coeff{i});
end
C=double(C);
position_zeros=reshape(Points_b',[],1);% Points_b gives
    simple boundary conditions
[row_posi_zeros,~]=size(position_zeros);
c_a=zeros(1,row_posi_zeros);%dimension of a complete
    displacement vector
%put coefficients into positions that are associated with
    this hinge
c_a((P1_number-1)*3+1:(P1_number-1)*3+3)=C(1:3);
c_a((P2_number-1)*3+1:(P2_number-1)*3+3)=C(4:6);
c_a((P3_number-1)*3+1:(P3_number-1)*3+3)=C(7:9);
c_a((P4_number-1)*3+1:(P4_number-1)*3+3)=C(10:12);
for j=row_posi_zeros:-1:1%remove DOFs that are
    constrained by boundaries
    if position_zeros(j)==1
        c_a(j)=[];
    end
end
c_a=real(c_a);

```

### A4. Identifying nodes that are associated with each hinge

```

function [hingeNodes_all,edge]=bars_hingeNodes_edgeZero(
    Points,Bars)
%This function produces 'four nodes order' which defines
    one hinge
%The order: hinges nodes first (1_top -> i and 2_bottom
    -> j), and then 3_left -> k and 4_right -> l
edge=[];
Rotation_90=[cos(pi/2),-sin(pi/2),0;sin(pi/2),cos(pi/2),
    ,0;0,0,1];
[row_bars,~]=size(Bars);
hingeNodes_all=[];
for i=1:row_bars
    [bars_adjacent, position_originPoint]=find(Bars==Bars(
        i,1));
    [row_bars_adjacent,~]=size(bars_adjacent);
    nodes_adjacent1=[];
    for j=1:row_bars_adjacent
        if position_originPoint(j)==1
            nodes_adjacent1=cat(2,nodes_adjacent1,Bars(
                bars_adjacent(j).2));
        else
            nodes_adjacent1=cat(2,nodes_adjacent1,Bars(
                bars_adjacent(j).1));
        end
    end
    [bars_adjacent, position_originPoint]=find(Bars==Bars(
        i,2));
    [row_bars_adjacent,~]=size(bars_adjacent);
    nodes_adjacent2=[];
    for j=1:row_bars_adjacent
        if position_originPoint(j)==1
            nodes_adjacent2=cat(2,nodes_adjacent2,Bars(
                bars_adjacent(j).2));
        else
            nodes_adjacent2=cat(2,nodes_adjacent2,Bars(
                bars_adjacent(j).1));
        end
    end
    hingeNodes_all=[];
    for i=1:row_bars
        [bars_adjacent, position_originPoint]=find(Bars==Bars(
            i,1));
        [row_bars_adjacent,~]=size(bars_adjacent);
        nodes_adjacent1=[];
        for j=1:row_bars_adjacent
            if position_originPoint(j)==1
                nodes_adjacent1=cat(2,nodes_adjacent1,Bars(
                    bars_adjacent(j).2));
            else
                nodes_adjacent1=cat(2,nodes_adjacent1,Bars(
                    bars_adjacent(j).1));
            end
        end
        [bars_adjacent, position_originPoint]=find(Bars==Bars(
            i,2));
        [row_bars_adjacent,~]=size(bars_adjacent);
        nodes_adjacent2=[];
        for j=1:row_bars_adjacent
            if position_originPoint(j)==1
                nodes_adjacent2=cat(2,nodes_adjacent2,Bars(
                    bars_adjacent(j).2));
            else
                nodes_adjacent2=cat(2,nodes_adjacent2,Bars(
                    bars_adjacent(j).1));
            end
        end
    end

```

#### A5. Computing fold angles and assembling compatibility matrix for all hinges

```
%% Fold angles for all hinges
[hingeNodes, edge.bars]=bars.hingeNodes.edgeZero(Points,
    Bars);
angles_1=[];
for i=1:number_hinges
    if hingeNodes(i,1)==0
        angleTemp=[];
    else
        angleP1=Points_1(hingeNodes(i,1),:);%x_i
        angleP2=Points_1(hingeNodes(i,2),:);%x_j
        angleP3=Points_1(hingeNodes(i,3),:);%x_k
        angleP4=Points_1(hingeNodes(i,4),:);%x_l
        angleTemp=config_fold_angle(angleP1,angleP2,angleP3,
            angleP4);
    end
    angles_1=[1, angles_1, angleTemp];
end
%% Compatibility matrix for all hinges
Matrix_B=[];
for i=1:number_hinges
    angleP1=hingeNodes(i,1);
    angleP2=hingeNodes(i,2);
    angleP3=hingeNodes(i,3);
    angleP4=hingeNodes(i,4);
    if angleP1==0
        Matrix_B_temp=[];
    else
        Matrix_B_temp=displacement_rotation(angleP1,angleP2,
            angleP3,angleP4,Points_1,all_info.Points_b);
    end
    Matrix_B=[1, Matrix_B, Matrix_B_temp];
end
```

#### A6. Proof of the convergence of correction steps

This subsection gives the proof for the convergence of the correction step (Section 3.5 and Check 2 in Fig. 7). Consider a mechanism consists of only pin-jointed bars as an example. It is sufficient to prove the convergence by proving that  $\|\hat{\mathbf{e}}_b^{i,1}\|$ , the norm of extensions of  $\mathbf{x}^{i,1}$ , is smaller than  $\|\hat{\mathbf{e}}_b^{i,0}\|$ , the norm of extensions of  $\mathbf{x}^{i,0}$ . The current extension-free configuration is  $\mathbf{x}^i$ , and a configuration that has second order extensions is given by Eq. (33) as

$$\mathbf{x}^{i,0} = \mathbf{x}^i + \alpha \mathbf{d}_m^i, \quad (87)$$

where  $\mathbf{d}_m^i$  is a unit vector. At  $\mathbf{x}^i$  and  $\mathbf{x}^{i,0}$ , the compatibility matrices of bars are respectively  $\mathbf{C}^b|_i$  and  $\mathbf{C}^b|_{i,0}$ . The complete extension of bar  $k$  at  $\mathbf{x}^{i,0}$  is given according to [23] as

$$\hat{\mathbf{e}}_b^k|_{i,0} = \mathbf{C}_k^b|_i \mathbf{d}_m^i \alpha + \frac{1}{2L_k} (\mathbf{d}_m^i)^T \mathbf{H}_k^b|_i \mathbf{d}_m^i |_k \alpha^2, \quad (88)$$

where

$$\mathbf{H}_k^b|_i = \tilde{\mathbf{H}} - (\mathbf{C}_k^b|_i \mathbf{C}_k^b|_i) \quad (89)$$

and

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathbf{I}_{3 \times 3} \\ -\mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}. \quad (90)$$

Because  $\mathbf{d}_m^i \in \text{Null}(\mathbf{C}^b|_i)$ ,

$$\hat{\mathbf{e}}_b^k|_{i,0} = \frac{1}{2L_k} (\mathbf{d}_m^i)^T \tilde{\mathbf{H}} \mathbf{d}_m^i |_k \alpha^2 = O(\alpha^2). \quad (91)$$

Putting together second order extensions of all bars gives

$$\hat{\mathbf{e}}_b^{i,0} = \begin{bmatrix} \vdots \\ \frac{1}{2L_k} (\mathbf{d}_m^i)^T \tilde{\mathbf{H}} \mathbf{d}_m^i |_k \alpha^2 \\ \vdots \end{bmatrix}. \quad (92)$$

Correction displacements are given by

$$\mathbf{C}^b|_{i,0} \mathbf{d}_c^{i,0} = -\hat{\mathbf{e}}_b^{i,0}. \quad (93)$$

It can be solved with Min.  $\|\mathbf{d}_c^{i,0}\|$  by SVD and it can be written in the form of pseudo-inverse as

$$\mathbf{d}_c^{i,0} = -(\mathbf{C}^b|_{i,0})^+ \hat{\mathbf{e}}_b^{i,0}. \quad (94)$$

$\mathbf{x}^{i,1}$  is given by

$$\mathbf{x}^{i,1} = \mathbf{x}^{i,0} + \mathbf{d}_c^{i,0}. \quad (95)$$

The corresponding extension of bar  $k$  at  $\mathbf{x}^{i,1}$  is

$$\hat{\mathbf{e}}_b^k|_{i,1} = \hat{\mathbf{e}}_b^k|_{i,0} + \mathbf{C}_k^b|_{i,0} \mathbf{d}_c^{i,0} + \frac{1}{2L_k} (\mathbf{d}_c^{i,0})^T \mathbf{H}_k^b|_i \mathbf{d}_c^{i,0} |_k. \quad (96)$$

Substitute Eqs. (91) and (94) into Eq. (96) and comparing to Eq. (91), it gives

$$\hat{\mathbf{e}}_b^k|_{i,1} = O(\alpha^4) < \hat{\mathbf{e}}_b^k|_{i,0} = O(\alpha^2), \quad (97)$$

when  $\alpha$  is sufficiently small. As a result,  $\|\hat{\mathbf{e}}_b^{i,1}\| < \|\hat{\mathbf{e}}_b^{i,0}\|$ , and the correction step is convergent.

#### A7. Some matrices and vectors in the example

This appendix provides further details on the results presented in Section 4.2.2. ‘Bars’ provides the nodal connection for all bars. For instance, bar 3 is connected by nodes 3 and 7. Some matrices and vectors are presented below, and some matrices are not in complete size due to the limited space. For path 1,



where only first 20 columns are presented.

$$\mathbf{W}_m|_0 = \left\{ \begin{array}{cccc} 3.6 \cdot 10^{-3} & 0.17 & 0.18 & -0.49 \\ 0.02 & -4.2 \cdot 10^{-3} & 0.079 & 0.35 \\ 0.61 & -0.018 & 0.047 & 0.02 \\ -0.37 & -0.053 & 0.32 & -0.15 \\ 4.4 \cdot 10^{-3} & 0.23 & 0.095 & -0.096 \\ 0.039 & -0.15 & 0.27 & 0.15 \\ -0.028 & -0.16 & -0.27 & 0.07 \\ 0.023 & 0.14 & 0.23 & -0.078 \\ 0.039 & -0.15 & 0.27 & 0.15 \\ 0.02 & -4.2 \cdot 10^{-3} & 0.079 & 0.35 \\ 0.019 & -0.098 & 0.15 & -0.019 \\ 1.8 \cdot 10^{-3} & 0.085 & 0.089 & -0.25 \\ 0.023 & 0.14 & 0.23 & -0.078 \\ 0.039 & -0.15 & 0.27 & 0.15 \\ -0.016 & -0.082 & -0.14 & 4.9 \cdot 10^{-3} \\ 0.023 & 0.14 & 0.23 & -0.078 \\ 6.9 \cdot 10^{-3} & -1.5 \cdot 10^{-3} & 0.028 & 0.12 \\ -0.022 & -0.12 & -0.21 & 0.037 \\ 0.012 & 0.071 & 0.12 & -0.022 \\ 0.019 & 0.44 & 0.17 & 0.2 \\ -3.1 \cdot 10^{-3} & 0.055 & -0.055 & 0.15 \\ 0.019 & -0.098 & 0.15 & -0.019 \\ -6.9 \cdot 10^{-3} & 1.5 \cdot 10^{-3} & -0.028 & -0.12 \\ 7.9 \cdot 10^{-3} & -1.7 \cdot 10^{-3} & 0.032 & 0.14 \\ 0.019 & -0.098 & 0.15 & -0.019 \\ -0.016 & 0.098 & -0.14 & 0.079 \\ 9.1 \cdot 10^{-3} & -0.056 & 0.079 & -0.046 \\ 6.9 \cdot 10^{-3} & -1.5 \cdot 10^{-3} & 0.028 & 0.12 \\ 7.9 \cdot 10^{-3} & -1.7 \cdot 10^{-3} & 0.032 & 0.14 \\ -3.5 \cdot 10^{-18} & -4.9 \cdot 10^{-17} & 1.6 \cdot 10^{-17} & -8.7 \cdot 10^{-18} \\ -3.4 \cdot 10^{-3} & 7.3 \cdot 10^{-4} & -0.014 & -0.06 \\ 2.0 \cdot 10^{-3} & -4.2 \cdot 10^{-4} & 7.9 \cdot 10^{-3} & 0.035 \\ 6.9 \cdot 10^{-3} & -1.5 \cdot 10^{-3} & 0.028 & 0.12 \\ -0.32 & -0.051 & 0.14 & 0.32 \\ 0.61 & -0.018 & 0.047 & 0.02 \\ -0.042 & 0.47 & -0.3 & 0.089 \\ 0.024 & -0.27 & 0.18 & -0.052 \\ 0.019 & 0.44 & 0.17 & 0.2 \end{array} \right\}, \quad \hat{\mathbf{d}}_m^0 = \left\{ \begin{array}{c} 3.6 \\ -7.4 \\ -3.9 \\ -3.6 \\ -0.64 \\ -8.8 \\ 5.4 \\ -4.3 \\ -8.8 \\ 0 \\ -7.4 \\ -3.3 \\ 1.8 \\ -4.3 \\ -8.8 \\ 3.3 \\ -4.3 \\ -2.6 \\ 4.3 \\ -2.5 \\ -7.1 \\ 0 \\ -1.0 \\ -3.3 \\ 2.6 \\ -3.0 \\ -3.3 \\ 2.1 \\ -1.2 \\ -2.6 \\ 0 \\ -3.0 \\ -2.3 \cdot 10^{-16} \\ 1.3 \\ -0.74 \\ -2.6 \\ 0 \\ 0 \\ 0 \\ -7.0 \\ -3.9 \\ 6.1 \\ -3.5 \\ -7.1 \end{array} \right\} \quad (101)$$

where removed DOF (boundary reactions) is added back to  $\hat{\mathbf{d}}_m^0$  as zeros.

$$\mathbf{x}^{0,0} = \begin{pmatrix} 0.06315 \\ -17.45 \\ 4.931 \\ 7.437 \\ -13.0 \\ -0.1555 \\ 7.594 \\ -13.07 \\ 4.845 \\ 0 \\ -8.79 \\ 4.941 \\ 7.532 \\ -13.07 \\ 4.845 \\ 7.558 \\ -4.405 \\ 4.955 \\ 15.08 \\ -8.704 \\ 4.875 \\ 0 \\ -8.678 \\ -0.05871 \\ 0.04495 \\ -8.712 \\ 4.941 \\ 7.536 \\ -4.351 \\ -0.04495 \\ 0 \\ 0 \\ -0.0519 \\ 5.0 \\ 7.522 \\ -4.343 \\ 4.955 \\ 0 \\ 0 \\ 0 \\ 0 \\ -17.44 \\ -0.06864 \\ 15.11 \\ -8.722 \\ -0.125 \end{pmatrix} = \begin{pmatrix} 0 \\ -17.32 \\ 5.00 \\ 7.50 \\ -12.99 \\ 0 \\ 7.50 \\ -12.99 \\ 5.00 \\ 0 \\ -8.66 \\ 5.00 \\ 7.50 \\ -12.99 \\ 5.00 \\ 7.50 \\ -4.33 \\ 5.00 \\ -8.66 \\ 5.00 \\ 0 \\ -8.66 \\ 0 \\ 0 \\ -8.66 \\ 0 \\ 0 \\ -8.66 \\ 5.00 \\ 7.50 \\ -4.33 \\ 0 \\ 0 \\ 0 \\ 0 \\ -17.32 \\ 0 \\ 15.00 \\ -8.66 \\ 0 \end{pmatrix} + 0.5 \begin{pmatrix} 0.13 \\ -0.26 \\ -0.14 \\ -0.13 \\ -0.022 \\ -0.31 \\ 0.19 \\ -0.15 \\ -0.31 \\ 0 \\ -0.26 \\ -0.12 \\ 0.063 \\ -0.15 \\ -0.31 \\ 0.12 \\ -0.15 \\ -0.09 \\ 0.15 \\ -0.088 \\ -0.25 \\ 0 \\ -0.036 \\ -0.12 \\ 0.09 \\ -0.1 \\ -0.12 \\ 0.072 \\ -0.042 \\ -0.09 \\ 0 \\ 0 \\ -0.1 \\ -8.2 \cdot 10^{-18} \\ 0.045 \\ -0.026 \\ -0.09 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.25 \\ -0.14 \\ 0.21 \\ -0.12 \\ -0.25 \end{pmatrix}. \quad (102)$$

$$\left( \begin{array}{c} \mathbf{e}_b^{0,0} \\ \mathbf{e}_c^{0,0} \end{array} \right) = \left\{ \begin{array}{c} 2.9 \cdot 10^{-4} \\ 2.6 \cdot 10^{-4} \\ -5.4 \cdot 10^{-4} \\ -6.1 \cdot 10^{-4} \\ 2.9 \cdot 10^{-4} \\ 1.9 \cdot 10^{-3} \\ 2.9 \cdot 10^{-3} \\ 1.3 \cdot 10^{-4} \\ 4.3 \cdot 10^{-4} \\ 2.5 \cdot 10^{-5} \\ 2.3 \cdot 10^{-4} \\ 2.3 \cdot 10^{-4} \\ 1.2 \cdot 10^{-3} \\ 2.4 \cdot 10^{-4} \\ 6.7 \cdot 10^{-4} \\ 1.3 \cdot 10^{-3} \\ 1.6 \cdot 10^{-3} \\ 7.8 \cdot 10^{-4} \\ 4.4 \cdot 10^{-4} \\ 3.4 \cdot 10^{-4} \\ 6.1 \cdot 10^{-5} \\ 1.3 \cdot 10^{-4} \\ 3.2 \cdot 10^{-4} \\ 3.2 \cdot 10^{-4} \\ 6.8 \cdot 10^{-4} \\ 7.7 \cdot 10^{-4} \\ 7.5 \cdot 10^{-4} \\ 5.2 \cdot 10^{-5} \\ 5.8 \cdot 10^{-4} \\ 1.3 \cdot 10^{-4} \\ 2.7 \cdot 10^{-4} \\ 6.6 \cdot 10^{-6} \\ 4.0 \cdot 10^{-4} \\ 6.4 \cdot 10^{-5} \\ 1.2 \cdot 10^{-4} \\ 1.9 \cdot 10^{-15} \\ 1.9 \cdot 10^{-15} \\ 3.9 \cdot 10^{-15} \\ 3.9 \cdot 10^{-15} \end{array} \right\}, \quad \mathbf{d}_c^{0,0} = \left\{ \begin{array}{c} 0.06344 \\ -17.45 \\ 4.931 \\ 7.436 \\ -13.0 \\ -0.1536 \\ 7.594 \\ -13.06 \\ 4.843 \\ 0 \\ 1.7 \cdot 10^{-5} \\ -8.79 \\ -8.4 \cdot 10^{-4} \\ -6.0 \cdot 10^{-4} \\ 4.9 \cdot 10^{-4} \\ 5.7 \cdot 10^{-4} \\ 1.2 \cdot 10^{-4} \\ -2.5 \cdot 10^{-4} \\ -2.7 \cdot 10^{-4} \\ -1.9 \cdot 10^{-4} \\ 1.1 \cdot 10^{-4} \\ -5.5 \cdot 10^{-4} \\ 0 \\ -2.2 \cdot 10^{-4} \\ 4.0 \cdot 10^{-4} \\ -8.9 \cdot 10^{-5} \\ 2.2 \cdot 10^{-4} \\ 8.7 \cdot 10^{-5} \\ -6.9 \cdot 10^{-5} \\ 4.0 \cdot 10^{-5} \\ 8.1 \cdot 10^{-5} \\ 0 \\ -9.1 \cdot 10^{-5} \\ -2.7 \cdot 10^{-4} \\ -1.6 \cdot 10^{-4} \\ 9.3 \cdot 10^{-5} \\ 5.5 \cdot 10^{-5} \\ 0 \\ -3.8 \cdot 10^{-4} \\ -1.4 \cdot 10^{-4} \\ 5.1 \cdot 10^{-4} \\ -2.9 \cdot 10^{-4} \\ -4.2 \cdot 10^{-4} \end{array} \right\}, \quad \mathbf{x}^{0,1} = \left\{ \begin{array}{c} 0.06344 \\ -17.45 \\ 4.931 \\ 7.436 \\ -13.0 \\ -0.1536 \\ 7.594 \\ -13.06 \\ 4.843 \\ 0 \\ -8.79 \\ 4.94 \\ -8.79 \\ 4.94 \\ 7.531 \\ -13.06 \\ 4.845 \\ 7.558 \\ -4.405 \\ 4.955 \\ 15.08 \\ -8.704 \\ 4.874 \\ 0 \\ -8.678 \\ -0.05831 \\ 0.04486 \\ -8.712 \\ 4.941 \\ 7.536 \\ -4.351 \\ -0.04487 \\ 0 \\ -0.05199 \\ 5.0 \\ 7.522 \\ -4.343 \\ 4.955 \\ 0 \\ -17.44 \\ -0.06878 \\ 15.11 \\ -8.722 \\ -0.1254 \end{array} \right\}, \quad \mathbf{x}^1 = \left\{ \begin{array}{c} 0.06344 \\ -17.45 \\ 4.931 \\ 7.436 \\ -13.0 \\ -0.1536 \\ 7.594 \\ -13.06 \\ 4.843 \\ 0 \\ -8.79 \\ 4.94 \\ -8.79 \\ 4.94 \\ 7.531 \\ -13.06 \\ 4.845 \\ 7.558 \\ -4.405 \\ 4.955 \\ 15.08 \\ -8.704 \\ 4.874 \\ 0 \\ -8.678 \\ -0.05831 \\ 0.04486 \\ -8.712 \\ 4.941 \\ 7.536 \\ -4.351 \\ -0.04487 \\ 0 \\ -0.05199 \\ 5.0 \\ 7.522 \\ -4.343 \\ 4.955 \\ 0 \\ -17.44 \\ -0.06878 \\ 15.11 \\ -8.722 \\ -0.1254 \end{array} \right\}. \quad (103)$$

For path 2,

$$\mathbf{C}^{rh}|_0 = \left( \begin{array}{cccccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.067 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.13 & -0.1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & -0.067 & 0 & 0 & 0 & 0 & 0 & 0 & -0.067 & 0 & 0 & 0.13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.067 & 0.1 & -0.17 & 0 & 0 & 0 & 0 & 0.13 & -0.1 & 0.17 & -0.067 & 0 & 0 & 0 & 0 & 0 \\ 0.058 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.17 & 0.1 & 0 & 0.12 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.23 & 0 & 0 & 0 & 0.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & 0 & 0 & 0.2 & 0 & -0.067 & 0 & 0 & 0 & 0 & 0 & 0 & -0.067 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.13 & 0 & 0 & 0 & 0 & -0.1 & -0.17 & -0.067 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0.058 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.12 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.058 & -0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.13 & 0 & 0 & 0 & 0 & 0 & 0 & -0.17 & -0.067 & 0 & 0 & -0.067 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & -0.067 & 0 & 0 & -0.067 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.13 & 0.1 & -0.17 & -0.067 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.067 & 0 & 0 & 0.13 & 0 & 0 & \dots \end{array} \right) \quad (105)$$

where first 18 columns are presented. For path 3,

$$\mathbf{C}^a|_0 = \left\{ \begin{array}{ccccccccccccccccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.87 & 0.5 & 0 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.87 & -0.5 & 0.87 & 0.5 & 0 & 0 \end{array} \right\}, \quad (106)$$

which is corresponding to hinge 12, 16, and 20.

## References

- [1] You Z. Motion structures extend their reach. *Mater Today* 2007;10(12):52–7. doi:[10.1016/S1369-7021\(07\)70308-5](https://doi.org/10.1016/S1369-7021(07)70308-5).
- [2] Miura K, Furuya H, Suzuki K. Variable geometry truss and its application to deployable truss and space crane arm. *Acta Astronaut* 1985;12(7):599–607. doi:[10.1016/0094-5765\(85\)90131-6](https://doi.org/10.1016/0094-5765(85)90131-6).
- [3] Overvelde JTB, Weaver JC, Hoberman C, Bertoldi K. Rational design of reconfigurable prismatic architected materials. *Nature* 2017;541:347–52. doi:[10.1038/nature20824](https://doi.org/10.1038/nature20824).
- [4] Clarricoats PJB, Zhou H. Design and performance of a reconfigurable mesh reflector antenna. 1. Antenna design. *IEE Proc H (Microw Antennas Propag)* 1991;138(6):485–92.
- [5] Sessions D, Cook A, Fuchi K, Gillman A, Huff G, Buskohl P. Origami-inspired frequency selective surface with fixed frequency response under folding. *Sensors (Basel)* 2019;19(21):4808. doi:[10.3390/s19214808](https://doi.org/10.3390/s19214808).
- [6] Keidel D, Molinari G, Ermanni P. Aero-structural optimization and analysis of a camber-morphing flying wing: structural and wind tunnel testing. *J Intell Mater Syst Struct* 2019;30(6):908–23. doi:[10.1177/1045389X19828501](https://doi.org/10.1177/1045389X19828501).
- [7] Jenett B, Calisch S, Cellucci D, Cramer N, Gershenfeld N, Swei S, et al. Digital morphing wing: Active wing shaping concept using composite lattice-based cellular structures. *Soft Robotics* 2017;4(1):33–48. doi:[10.1089/soro.2016.0032](https://doi.org/10.1089/soro.2016.0032). PMID: 28289574
- [8] You Z, Pellegrino S. Foldable bar structures. *Int J Solids Struct* 1997;34(15):1825–47. doi:[10.1016/S0020-7683\(96\)00125-4](https://doi.org/10.1016/S0020-7683(96)00125-4).
- [9] Kwan A, Pellegrino S. A new concept for large deployable space frames. *Int J Space Struct* 1994;9(3):153–62. doi:[10.1177/026635119400900304](https://doi.org/10.1177/026635119400900304).
- [10] Nishiyama Y. Miura folding: applying origami to space exploration. *Int J Pure Appl Math* 2012;79:269–79.
- [11] Kurabayashi K, Tsuchiya K, You Z, Tomus D, Umemoto M, Ito T, et al. Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Mater Sci Eng* 2006;419(1):131–7. doi:[10.1016/j.msea.2005.12.016](https://doi.org/10.1016/j.msea.2005.12.016).
- [12] Onal CD, Wood RJ, Rus D. An origami-inspired approach to worm robots. *IEEE/ASME Trans Mechatron* 2013;18(2):430–8. doi:[10.1109/TMECH.2012.2210239](https://doi.org/10.1109/TMECH.2012.2210239).
- [13] Hernandez P, Edwin, Hartl, Darren, Malak, Richard, et al. Origami-inspired active structures: a synthesis and review. *Smart Mater Struct* 2014;23:94001. doi:[10.1088/0964-1726/23/9/094001](https://doi.org/10.1088/0964-1726/23/9/094001).
- [14] McCarthy J. Geometric design of linkages. *Interdisciplinary Applied Mathematics*. Springer New York; 2000.
- [15] Gattas JM, Wu W, You Z. Miura-base rigid origami: parameterizations of first-level derivative and piecewise geometries. *J Mech Des* 2013;135(11):111011. doi:[10.1115/1.4025380](https://doi.org/10.1115/1.4025380).
- [16] Overvelde JTB, de Jong TA, Shevchenko Y, Becerra SA, Whiteside G, Weaver JC, et al. A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees of freedom. *Nat Commun* 2016;7:10929. doi:[10.1038/ncomms10929](https://doi.org/10.1038/ncomms10929).
- [17] Lubbers LA, van Hecke M. Excess floppy modes and multibranched mechanisms in metamaterials with symmetries. *Phys Rev E* 2019;100(2). doi:[10.1103/physreve.100.021001](https://doi.org/10.1103/physreve.100.021001).
- [18] Coulais C, Sabbadini A, Vink F, van Hecke M. Multi-step self-guided pathways for shape-changing metamaterials. *Nature* 2018;561:512–15. doi:[10.1038/s41586-018-0541-0](https://doi.org/10.1038/s41586-018-0541-0).
- [19] Gatt R, Mizzi L, Azzopardi JI, Azzopardi KM, Attard D, Casha A, et al. Hierarchical auxetic mechanical metamaterials. *Sci Rep* 2017;5:8395. doi:[10.1038/srep08395](https://doi.org/10.1038/srep08395).
- [20] Yakey JH, LaValle SM, Kavraki LE. Randomized path planning for linkages with closed kinematic chains. *IEEE Trans Robot Autom* 2001;17(6):951–8. doi:[10.1109/70.976030](https://doi.org/10.1109/70.976030).
- [21] Trinkle J, Milgram RJ. Complete path planning for closed kinematic chains with spherical joints. *Int J Robot Res* 2002;21(9):773–89. doi:[10.1177/027836490221009119](https://doi.org/10.1177/027836490221009119).
- [22] Magliozzi Luca, Micheletti Andrea, Pizzigoni Attilio, Ruscica Giuseppe. On the design of origami structures with a continuum of equilibrium shapes. *Composites Part B* 2017;115:144–50 ISSN 1359-8368. In this issue. doi:[10.1016/j.compositesb.2016.10.023](https://doi.org/10.1016/j.compositesb.2016.10.023).
- [23] Li Y, Pellegrino S. A theory for the design of multi-stable morphing structures. *J Mech Phys Solids* 2019;103772. doi:[10.1016/j.jmps.2019.103772](https://doi.org/10.1016/j.jmps.2019.103772).
- [24] Lang RJ, Magleby S, Howell L. Single degree-of-freedom rigidly foldable cut origami flashers. *Journal of Mechanisms and Robotics* 2016;8(3). doi:[10.1115/1.4032102](https://doi.org/10.1115/1.4032102). 031005
- [25] Chen Y, You Z, Tarnai T. Threefold-symmetric Bricard linkages for deployable structures. *Int J Solids Struct* 2005;42(8):2287–301. doi:[10.1016/j.ijsolstr.2004.09.014](https://doi.org/10.1016/j.ijsolstr.2004.09.014).
- [26] Lang R. *Twists, tilings, and tessellations*. A K Peters, Limited; 2017.
- [27] Liu K, Paulino GH. Nonlinear mechanics of non-rigid origami: an efficient computational approach. *Proc R Soc Lond A* 2017;473(2206). doi:[10.1098/rspa.2017.0348](https://doi.org/10.1098/rspa.2017.0348).
- [28] Tachi T. Simulation of rigid origami. *Origami 4: Fourth international meeting of origami science, mathematics, and education*, 4; 2009. doi:[10.1201/b10653-20](https://doi.org/10.1201/b10653-20).
- [29] Tachi T. Designing freeform origami tessellations by generalizing Resch's patterns. *Journal of Mechanical Design* 2013;135(11). doi:[10.1115/1.4025389](https://doi.org/10.1115/1.4025389). 111006
- [30] Xi Z, Lien J. Folding rigid origami with closure constraints. In: 38th Mechanisms and robotics conference, international design engineering technical conferences and computers and information in engineering conference, Volume 5B; 2014 <https://doi.org/10.1115/DETC2014-35556>.
- [31] Xi Z, Lien J. Plan folding motion for rigid self-folding machine via discrete domain sampling. In: 2015 IEEE International conference on robotics and automation (ICRA); 2015. p. 2938–43. doi:[10.1109/ICRA.2015.7139601](https://doi.org/10.1109/ICRA.2015.7139601).
- [32] Kumar P, Pellegrino S. Computation of kinematic paths and bifurcation points. *Int J Solids Struct* 2000;37(46):7003–27. doi:[10.1016/S0020-7683\(99\)00327-3](https://doi.org/10.1016/S0020-7683(99)00327-3).
- [33] Zhang T, Kawaguchi K, Wu M. A folding analysis method for origami based on the frame with kinematic indeterminacy. *Int J Mech Sci* 2018;146–147:234–48. doi:[10.1016/j.ijmecsci.2018.07.036](https://doi.org/10.1016/j.ijmecsci.2018.07.036).
- [34] Pellegrino S, Calladine C. Matrix analysis of statically and kinematically indeterminate frameworks. *Int J Solids Struct* 1986;22(4):409–28. doi:[10.1016/0020-7683\(86\)90014-4](https://doi.org/10.1016/0020-7683(86)90014-4).
- [35] Gottwald S. *The VNR concise encyclopedia of mathematics*. Springer Netherlands; 2012.
- [36] Pellegrino S. Structural computations with the singular value decomposition of the equilibrium matrix. *Int J Solids Struct* 1993;30(21):3025–35. doi:[10.1016/0020-7683\(93\)90210-X](https://doi.org/10.1016/0020-7683(93)90210-X).
- [37] Lang R. *Tessellatica*. 2017b. URL <http://www.langorigami.com/article/tessellatica>.
- [38] Wikipedia contributors. Principal component analysis. 2019. [Online; accessed 13-October-2019]; URL [https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis).
- [39] Jnsson H, Mills G, Jacobsen K.W.. Nudged elastic band method for finding minimum energy paths of transitions. p. 385–404.. [https://www.worldscientific.com/doi/pdf/10.1142/9789812839664\\_0016](https://www.worldscientific.com/doi/pdf/10.1142/9789812839664_0016); URL [https://www.worldscientific.com/doi/abs/10.1142/9789812839664\\_0016](https://www.worldscientific.com/doi/abs/10.1142/9789812839664_0016). 10.1142/9789812839664\_0016