

Computational Design of Planar Multistable Compliant Structures

RAN ZHANG, IST Austria
THOMAS AUZINGER, IST Austria
BERND BICKEL, IST Austria



Fig. 1. The optimization-based design tool that we have introduced enables the computation of multistable compliant structures that can, e.g., be used to pose physical figurines.

This paper presents a method for designing planar multistable compliant structures. Given a sequence of desired stable states and the corresponding poses of the structure, we identify the topology and geometric realization of a mechanism—consisting of bars and joints—that is able to physically reproduce the desired multistable behavior. In order to solve this problem efficiently, we build on insights from minimally rigid graph theory to identify simple but effective topologies for the mechanism. We then optimize its geometric parameters, such as joint positions and bar lengths, to obtain correct transitions between the given poses. Simultaneously, we ensure adequate stability of each pose based on an effective approximate error metric related to the elastic energy Hessian of the bars in the mechanism. As demonstrated by our results, we obtain functional multistable mechanisms of manageable complexity that can be fabricated using 3D printing. Further, we evaluated the effectiveness of our method on a large number of examples in the simulation and fabricated several physical prototypes.

CCS Concepts: • **Applied computing** → **Computer-aided manufacturing**; *Computer-aided design*; • **Mathematics of computing** → **Graph enumeration**; • **Theory of computation** → *Computational geometry*;

Additional Key Words and Phrases: multistability, mechanism, computational design, rigidity

Authors' addresses: Ran Zhang, IST Austria, Am Campus 1, 3400, Klosterneuburg, rzhang@ist.ac.at; Thomas Auzinger, IST Austria, Am Campus 1, 3400, Klosterneuburg, thomas.auzinger@ist.ac.at; Bernd Bickel, IST Austria, Am Campus 1, 3400, Klosterneuburg, bernd.bickel@ist.ac.at.

© 2021 Copyright held by the owner/author(s).
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, <https://doi.org/10.1145/3453477>.

ACM Reference Format:

Ran Zhang, Thomas Auzinger, and Bernd Bickel. 2021. Computational Design of Planar Multistable Compliant Structures. *ACM Trans. Graph.* 1, 1, Article 1 (January 2021), 16 pages. <https://doi.org/10.1145/3453477>

1 INTRODUCTION

Striking an iconic pose can provide an extremely strong visual impression, and several simple iconic poses have solidified their place in history. Famous examples are Michael Jordan's "Jumpman", a prominent brand logo for basketball sneakers, victory poses of athletes such as Usain Bolt or Tiger Woods, or the dance moves of various performers.

Motivated by the advancement of computer-controlled manufacturing devices, in recent years we have witnessed impressive novel tools emerging for the design of articulated characters and personalized automata and animatronics—for example, by designing fabricable joints [Bächer et al. 2012], gears [Coros et al. 2013], linkage systems [Bächer et al. 2015], or kinetic wires [Xu et al. 2018] to cover a specific *motion range* or reproduce a desired *movement*. In contrast, in our work, we are primarily interested in reproducing and being able to transition between a small, specific set of *discrete poses* supported by a mechanical system. From a practical viewpoint, when transitioning between poses, the mechanism should guide or ideally snap to one of the prescribed poses, thereby ensuring that the desired pose is correctly reproduced. Further, a character must be able to hold the pose in a stable manner, even in the presence of external disturbances—for example, when being touched or lifted. From a mechanical viewpoint, this desirable property is described by the concept of multistability.

By definition, a multistable structure is a structure that permits multiple stable states. Each state is characterized as a local minimum in the potential energy landscape of the structure. Important mechanical characteristics are reflected by the mutual reachability of states (i.e., is there a feasible transition between them) and the stability of each state (i.e., how much energy is required to initiate such transitions). Multistability is a fundamental property for transforming a device between desired states [Chen et al. 2017] and has received considerable interest in the engineering community for the design of robotic manipulators [Wingert et al. 2002], active building facades [Golabchi and Guest 2009], large-scale space structures [Schioler and Pellegrino 2007], energy-absorbing elements [Leelavanichkul et al. 2010], and auxetic metamaterials [Rafsanjani and Pasini 2016]. However, most of these structures are designed by hand and/or composed of an assembly of parameterized template units, which are usually bistable mechanisms with a low number of degrees of freedom [Lumpe and Shea 2019]. This makes it challenging to accurately reproduce an arbitrary user-given set of states, can lead to an unnecessarily large number of units, and—as their complexity increases—makes it difficult to explore their design space and customize them to user-specified needs. Therefore, an important research question is how to identify new multistable mechanisms and adapt them to new applications.

In this paper, we propose a novel computational approach for designing layered *multistable compliant* structures. Our structures are composed of linkages that comprise hinge joints and compliant as well as rigid bars. We propose an optimization-based method that jointly considers and optimizes topology, position of stable states, rigidity, and fabricability. Our approach is based on insights from minimally rigid graph theory, which enables us to efficiently identify feasible configurations. As demonstrated by our results, we obtain mechanisms with a small number of components that can be easily fabricated using 3D printing. We evaluated our method with a large number of examples in simulation and fabricated and tested several real-world models.

2 RELATED WORK

In recent years, computer graphics research has significantly contributed to new methods and representations for fabrication-aware design [Bermano et al. 2017; Bickel et al. 2018], also involving the investigation of a broad range of applications related to mechanical systems and structures. For example, this includes the design of deployable structures [Konaković-Luković et al. 2018], structured materials [Martínez et al. 2016; Panetta et al. 2015], structurally optimized shapes using topology optimization [Liu et al. 2018; Zhu et al. 2017] with user-controllable geometric features [Dumas et al. 2015], and linkage systems [Bächer et al. 2015]. Building on a long tradition of virtual character modeling and animation, a particular challenge in this context are methods for synthesizing or retargeting 3D printable mechanisms whose motions resemble those of virtual characters [Zhang et al. 2017]. Ceylan et al. [2013] compute mechanical automata from mocap sequences, while Coros et al. [2013] use user-specified motion-curves as input. Thomaszewski et al. [2014] begin with a fully-actuated mechanical structure that performs a certain periodic motion, and then iteratively replace joint motors

with rigid links to obtain linkage-based characters. Using as input a conventional, rigidly-actuated mechanism, Megaro et al. [2017] proposed a computational tool to replace joints with parameterized flexures, thereby enabling to efficiently turn the input into a compliant mechanism. To ease defining the motion, Megaro et al. [2014] proposed a sketch-based modeling approach for two-dimensional mechanical characters through the placement of extreme poses. Along this line, beginning with a few key poses of multiple moving parts, Nishida et al. [2019] proposed a computational design system for interactively constructing mechanical linkages that translate moving elements through discrete target configurations. In contrast to these works that focus on reproducing a specific *motion* with a periodic actuation loop, our work aims at designing a multistable compliant structure that exhibits a set of *discrete* poses that are capable of steadily remaining at these distinct positions without external force or an additional locking mechanism. Moreover, these works typically use fixed linkage topologies or combine well-defined elementary linkage topologies, such as four-bar linkages, Theo-Jansen linkages, and Klann linkages. In contrast, we do not prescribe fixed topologies but design our multistable bar-joint mechanism by taking into account a much more general class of suitable graphs (i.e., minimally rigid graphs). Thus, we not only optimize the vertex positions of such graphs, but also the graph topology itself. This aspect of our work might be relevant for conventional linkage design as well.

More similar to our work, Pan et al. [2019] identified globally optimal topologies and realizations for planar linkages by solving the corresponding mixed-integer convex optimization problem. In their approach, topological correctness to ensure symbolic computability and that the linkage structure contains no wasted structures is enforced with a set of constraints. While applicable to linkages of low complexity with a single driver and end effector, we handle multiple end effectors, including the construction of a suitable driver. We identify feasible topologies using simple construction rules from rigid graph theory and thereby avoid the need of complex and often intractable topology optimizations on graphs.

2.1 Multistable Mechanisms

Multistable mechanisms and their design are the foundation for a plethora of exciting applications. Facilitating the creation of programmable multistable surfaces, Bende et al. [2015] introduced design rules using creases and curvature to generate snapping transitions in arbitrary surfaces. Bharaj et al. [2018] suggested a computational design method that can create a fabricable planar structure that can morph into two different bistable forms, resembling the outlines of two shapes. Ion et al. [2017] presented a workflow for designing a functional-complete mechanical logic system with embedded bistable elements, which enables signals to propagate and amplify through bi-stable springs. Utilizing multistable micro-flexures, Song et al. [2019] performed Boolean computations based purely on mechanical forces and displacements. A key benefit of these systems is that they can be additively fabricated and execute their intended function without requiring electric power. For design, these systems rely on templates of bi-stable or multistable mechanisms with predefined topology provided as input. The design space

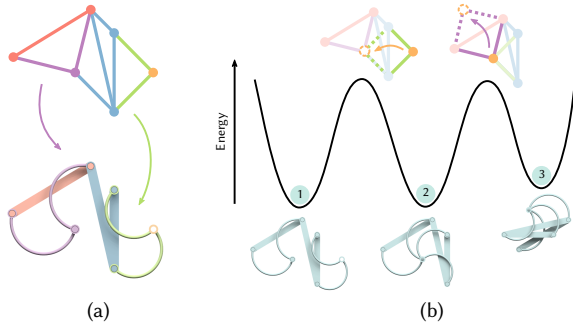


Fig. 2. Multistability of a bar-joint mechanism. (a) A multistable bar-joint mechanism can be constructed from a minimally rigid graph by setting certain graph edges (green and violet) as compliant bars (the reference edge in red is kept fixed). (b) Multiple realizations of the minimally rigid graph yield local minima in the energy landscape of the mechanism. Different stable states can be achieved by crossing over finite energy barriers that result from the bending of compliant bars.

and performance of compliant bistable micromechanisms were studied using finite element analysis [Jensen et al. 1999], and Chen et al. [2011] synthesized compliant multistable mechanisms through the use of a single bistable mechanism. By applying local actuation patterns, Inguez-Rabago et al. [2019] explored the folding behavior of a range of 3D prismatic building blocks that exhibit controllable multifunctionality and visualize their complex mechanical behavior. In contrast, we propose a joint topology and geometric optimization of multistable mechanisms that does not rely on pre-designed building blocks. Using elastic and rigid bars connected by joints as a design space, our method enables the discovery of novel functional mechanisms with a low number of components.

3 OVERVIEW

Given a sequence of user-defined stable poses of an object, our goal is to computationally design a fabricable mechanism that realizes this behavior. Stated in such generality, this problem spans a vast design space. To make it tractable, we will first identify our design requirements.

3.1 Design Considerations

Mechanical Design. We assume that the user-provided model is a system consisting of rigid-body parts, some of which move between different user-defined poses. To enable this behavior, we design a multistable planar mechanical linkage, where a set of rigid and flexible bodies (i.e., the *links*) are connected by joints. In our case, we design a bar-and-joint system, as illustrated in Figure 2 (a), that consists—apart from the rigid parts of the user-provided model—of rigid and compliant *bars* as well as *revolute joints*.

Similar structures have been employed for the design of flexible linkages with revolute and compliant joints [Bächer et al. 2015; Megaro et al. 2017]. While these works designed one-degree-of-freedom linkages that aim to realize the motion curves of one or more end effectors, our setting exhibits fundamental differences. For multistability, it is not sufficient to identify a mechanism that passes through all user-specified poses; each of the poses must lie at

a local minimum of the internal energy introduced by the bending of compliant bars in the structure.

Structural Multistability. The notion of *multistability* plays a key role in our method. It reflects three essential properties of a valid multistable bar-joint mechanism. First, there are feasible transitions between the stable poses. Second, the stable poses must be locally stable in the mixed rigid-compliant bar-joint mechanism. Third, the mixed rigid-compliant bar-joint mechanism should not exhibit transitions into undesired stable poses.

While the first requirement is a traditional problem in linkage design, the latter two are not well formulated in previous research and it is not trivial to find a configuration that satisfies these two requirements. As elaborated in Section 4, we employ tools associated with graph rigidity theory to analyze this problem and develop efficient algorithms to design such mechanisms.

Fabricability. In order to ensure relevance for a wide user base, we do not require expertise in engineering or manufacturing when using our system or for the subsequent fabrication of the structure. Given the accessibility of modern additive manufacturing methods, we meet this requirement by generating structures that can be realized entirely on a commercial 3D printer with only minimal post-processing, like support removal. Simultaneously, we expect our designs to be directly handled by the user and forgo alternative drivers and power sources, such as motors or electromagnets.

Prioritizing convenient fabricability also influences the possible types of energy that can be utilized to ensure the stability of each user-specified pose. Printing magnetic structures is still in an experimental phase [Huber et al. 2016], while pneumatic and hydraulic structures require intricate geometries to make them both reliable and predictable pressure tanks [Ma et al. 2017]. Further, since gravitational forces are generally too weak, we employ the elastic energy of deformable structures.

3.2 Our Approach

Based on the aforementioned design considerations, we propose an automatic computational design system for the creation of multistable objects that assume user-specified poses. As illustrated in Figure 3, we take an object with multiple user-provided poses as input and aim to design a bar-joint mechanism that correctly *and stably* positions the end effector in each pose. This requires the joint optimization of (i) the overall topology of the mechanism (i.e., the number of joints and bars as well as their connectivity) and (ii) its actual geometric realization (i.e., the joint coordinates and bar lengths).

Using rigidity theory as the theoretical background (see Section 4), we explore feasible mechanism topologies and optimize their geometric specification to both match the desired poses and ensure the stability of each pose (see Section 5). As an immediate benefit, this theory allows us to avoid overconstrained graph topologies that would unnecessarily increase mechanism complexity. Moreover, we perform the exploration in a manner that prioritizes mechanism with a few bars and joints. This reduces the overall complexity of the final design and facilitates the creation of fabrication-ready models. Finally, we resolve potential collisions between parts and

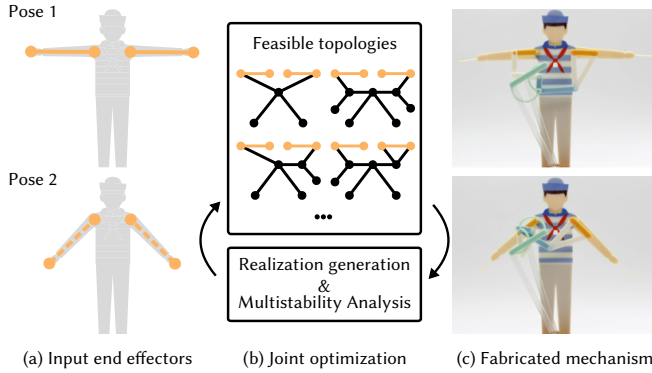


Fig. 3. Overview. Our system takes multiple poses of a rigid-body system as input and generates a fabrication-ready multistable bar-joint mechanism that realizes them. (a) First, in the SAILOR poses, the end effectors (in orange) are converted to joints and bars and form part of the final mechanism, which is (b) obtained by our automatic topological-geometric optimization, which explores feasible structures of the mechanism and analyzes the stability of optimized positions of the joints and bars. Having identified a mechanism that realizes the desired behavior, we (c) resolve potential collision issues and automatically generate 3D geometry for fabrication. After manual assembly of the 3D printed parts, the mechanism exhibits the desired target poses.

automatically generate 3D models for each part of the mechanism (see Section 6).

4 STRUCTURAL MULTISTABILITY

From a geometric viewpoint, the structural multistability of a bar-joint mechanism is determined by the rigidity of its graph representation. A bar-joint mechanism is often abstracted by its corresponding *undirected graph* $G = (V, E)$, where joints are represented by the vertices V of the graph and bars by its edges E . The actual geometric layout is given by a *realization function* $\rho : V \rightarrow \mathbb{R}^2$, which assigns a position $\rho(v)$ to each joint $v \in V$. Additionally, the geometric layout provides a labeling $\lambda : E \rightarrow \mathbb{R}_{>0}$ of the graph edges with lengths $\lambda(e)$.

This abstraction is also applicable for our setting, as it is possible to infer relevant aspects regarding multistability from it. The existence and number of stable poses is largely determined by the topology of the graph itself (see Sections 4.1 and 4.2). A quantitative measure of a pose’s stability can be obtained by analyzing the potential elastic energy of such graphs (see Section 4.3).

4.1 Topological Rigidity

Instead of specifying a concrete realization ρ for an undirected graph, one can also specify a labeling $\lambda : E \rightarrow \mathbb{R}_{>0}$ of the graph edges with lengths $\lambda(e)$. In effect, this fixes the length of each bar $e \in E$ of the mechanism. A central topic in the rigidity theory of graphs concerns itself with the number of realizations that are permitted by a graph G together with a set of edge lengths λ . For this, we call the pair (G, λ) a *framework* (see Figure 4). As depicted in Figure 5, a framework falls into one of several categories based on how many *compatible* realizations it permits. With *compatible*, we refer to realizations ρ , such that the assigned vertex

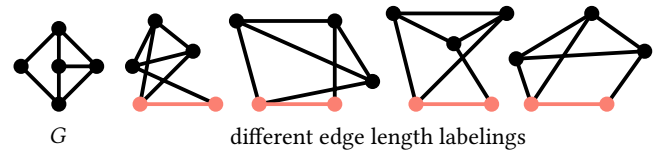


Fig. 4. Frameworks with varying edge length labelings. While the connectivity of a mechanism is determined by its connectivity graph G , its geometry is defined by the edge length labeling λ of its edges. A possible realization for each of the different edge length labelings is depicted.

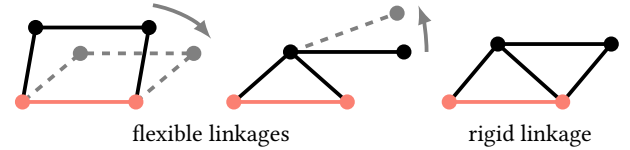


Fig. 5. *Flexible* linkages enable a continuous range of motion, even if one of their links (represented with a colored line) is kept fixed. Since we are interested in multistable poses, we consider *rigid* systems, which permit only a discrete set of configurations.

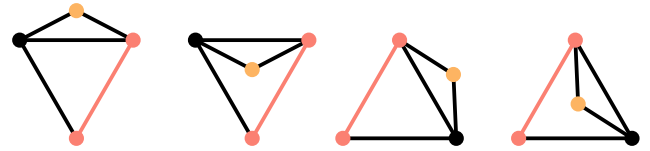


Fig. 6. Different realizations of the same rigid framework. By keeping the location of the colored bar fixed, the four different realization are shown. Note that the colored end effector vertex assumes a different location for each of the stable poses.

positions $\rho(v_i), \rho(v_j)$ conform to the specified edge lengths,

$$\lambda(e) = \|\rho(v_i) - \rho(v_j)\|, \quad \text{where } e = \{v_i, v_j\}. \quad (1)$$

If an edge labeling permits at least one realization, we call the corresponding framework *realizable*. A simple example that is *not* realizable are edge lengths for a triangle graph that do not satisfy the triangle inequality. For realizable frameworks, a rigid transformation of the entire graph, including translations and rotations, would again result in a valid realization. Thus, we consider two realization *equivalent* if they differ only by a rigid transformation. This can be avoided by fixing the location and orientation of one of the mechanism’s bars in all realizations that are considered (see colored bars in Figures 4 to 6). All our mechanisms are fixed in world space, and in the following account, only non-equivalent realizations are considered.

If a framework permits an infinitely large number of realizations, we call it *flexible* and two such examples are illustrated in Figure 5. This is the conventional case for previous works on linkage design, where usually one-degree-of-freedom mechanisms were considered [Bächer et al. 2015]. The four-bar linkage serves as a prominent example that enables a continuous range of motion. In contrast, we are primarily interested in *rigid* frameworks for which only a finite number of realizations exist (see Figure 5, right). A framework that

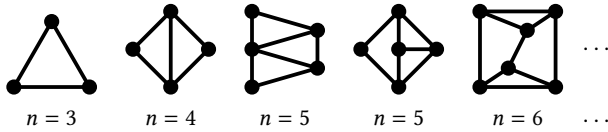


Fig. 7. Laman graphs with different vertex counts n . As generally *minimally rigid* graphs, they become flexible if any one edge is removed. Note that for each $n \geq 5$, multiple different Laman graphs exist.

forms a triangle, for example, can be mirrored along its fixed edge to form a simple bistable switch; with a combination of two such switches depicted in Figure 6.

Interestingly, the concept of rigidity can be extended to the topology of graphs, *without* referring to a particular edge length labeling. Omitting various technicalities—for which we refer to the recent work by Capco et al. [2018]—it is possible to define the notion of *generally rigid* graphs. For these graphs, almost all edge labelings lead to a finite number of possible realizations. Additionally, graphs can exhibit varying degrees of rigidity, ranging from (generally) *minimally rigid* graphs, which become flexible when any one of their edges is removed, to (generally) *globally rigid* graphs, which permit only a single realization (and its mirror image along the fixed edge). The former are of special interest to our setting, as detailed in the following section.

4.2 Laman Graphs

In two dimensions, generally minimally rigid graphs were first characterized by Hilda Pollaczek-Geiringer [1927]. Nevertheless, they were named after Gerard Laman, who rediscovered them decades later [Laman 1970]. As illustrated in Figure 7, a Laman graph on n vertices has exactly $2n - 3$ edges and every one of its k -vertex subgraphs has at most $2k - 3$ edges. Intuitively, this can be understood by considering the possible positions of the n vertices in 2D, yielding $2n$ degrees of freedom. Since we require rigidity, only the three degrees of freedom of a rigid transformation, which places the entire graph, are permitted. Each edge—with its corresponding edge length label—fixes the distance between its vertices and, accordingly, removes one degree of freedom from the vertex positions. Thus, exactly $2n - 3$ edges are required for a sufficient reduction of the degrees of freedoms. Simultaneously, no edge should be “wasted” by increasing the rigidity of an already rigid subgraph, thereby motivating the restriction on the subgraphs.

For our purposes, Laman graphs exhibit various convenient properties:

Simplicity For a given number of vertices, Laman graphs require the smallest number of edges to generate rigid frameworks. This makes them well suited for fabrication tasks, where unnecessary complexity negatively influences the construction time and introduces additional points of failure.

Practicability As discovered by Lebrecht Henneberg [1908], *all* Laman graphs can be constructed from a single edge by utilizing the following two operations: (i) adding a vertex and connecting it with two edges to existing vertices; and (ii) splitting an existing edge by inserting a new vertex, which is then

connected with a new edge to an additional existing vertex. Commonly called *Henneberg type I* and *Henneberg type II* steps, they enable an efficient construction and enumeration of Laman graphs.

Diversity Among all rigid graphs with a fixed number of vertices, Laman graphs allow for the largest number of possible realizations. In general, a Laman graph with n vertices exhibits at least 2^{n-2} realizations [Capco et al. 2018]. This allows us to design multistable frameworks using graphs of manageable complexity.

Symbolic Computability We rely on symbolic kinematics for efficient optimization. This requires that the mechanism can be processed sequentially, such that only one unknown joint position has to be determined at a time. While this is not the case for all planar linkages [Bächer et al. 2015], following the Henneberg type I construction rule guarantees the existence of such a processing sequence and, therefore, symbolic computability. In contrast, if the Henneberg type II rule is required for the construction of the graph, sequential processing is not possible. Based on this insight, we can conveniently restrict the considered topologies to a symbolically kinematically computable subset.

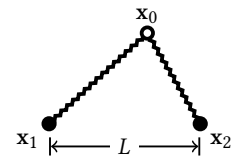
Further, Laman graphs can be converted to 2 DOF linkages, capable of transforming a 2D input motion to an output motion. These properties make Laman graphs applicable for effectively discovering the topologies for multistable mechanisms.

4.3 Physical Rigidity of Stable Poses

While the usage of Laman graphs ensures the overall rigidity of the mechanism in a graph-theoretic sense, we also require physical rigidity of each stable pose—that is, that the mechanism consisting of compliant bars can resist small forces that would push it out of its stable pose.

This design challenge could be formulated as a full-blown inverse physics-based simulation problem. As an efficient alternative, we propose a simple geometric criterion. It is inspired by the observation that our mechanisms can be constructed sequentially by applying the Henneberg type I construction rule. As explained in more detail in Section 5, we use it for inverse structure design to instantiate and optimize realizations of Laman graphs.

We consider a Henneberg type I construction step, which adds a vertex \mathbf{x}_0 and connects it to two existing vertices \mathbf{x}_1 and \mathbf{x}_2 as shown on the right. For our analysis and in order to position this bistable mechanism in space, we assume that \mathbf{x}_1 and \mathbf{x}_2 are fixed in space.



Without loss of generality, we assume the endpoints $\mathbf{x}_1 = (-L/2, 0)$ and $\mathbf{x}_2 = (L/2, 0)$, with distance L between them; the remaining vertex is located at $\mathbf{x}_0 = (x_0, y_0)$.

For the bars connecting \mathbf{x}_0 with \mathbf{x}_1 and \mathbf{x}_2 , we consider a linear elastic material with Young’s Modulus E and constant cross-sectional area A . Denoting the rest lengths of the two non-fixed edges with $d_{01} = \|\mathbf{x}_0 - \mathbf{x}_1\|$, $d_{02} = \|\mathbf{x}_0 - \mathbf{x}_2\|$, and the deformed lengths with $d_{01}^* = \|\mathbf{x}_0 + \mathbf{x} - \mathbf{x}_1\|$, $d_{02}^* = \|\mathbf{x}_0 + \mathbf{x} - \mathbf{x}_2\|$ we obtain

their elastic potential energy as

$$U(\mathbf{x}_0, \mathbf{x}) = \frac{k_0}{2d_{01}} (d_{01}^* - d_{01})^2 + \frac{k_0}{2d_{02}} (d_{02}^* - d_{02})^2,$$

where $\mathbf{x} = (x, y)$ denotes the displacement of the non-fixed vertex from \mathbf{x}_0 to $\mathbf{x}_0 + \mathbf{x}$.

The spring constants can be calculated by $k_0 = \frac{EA}{d_{ij}}$.

Without a displacement, i.e., $\mathbf{x} = \mathbf{0}$, we not only have $U(\mathbf{x}_0, \mathbf{0}) = 0$ but also $\nabla U(\mathbf{x}_0, \mathbf{0}) = \mathbf{0}$, since the non-fixed vertex is located in a stable pose at a minimum of the potential energy. In order to measure the rigidity of a stable pose, we use the Hessian \mathbf{H}_U of the potential energy, i.e.,

$$\begin{aligned} \mathbf{H}_U|_{\mathbf{x}=\mathbf{0}} &= \begin{pmatrix} \frac{\partial^2 U}{\partial x^2} & \frac{\partial^2 U}{\partial x \partial y} \\ \frac{\partial^2 U}{\partial y \partial x} & \frac{\partial^2 U}{\partial y^2} \end{pmatrix} (\mathbf{x}_0, \mathbf{0}) = \\ &= k_0 \begin{pmatrix} \sigma_1 - y_0^2 \sigma_3 & y_0 (x_0 \sigma_3 + L \Delta_3) \\ y_0 (x_0 \sigma_3 + L \Delta_3) & y_0^2 \sigma_3 \end{pmatrix} \end{aligned}$$

with $\sigma_1 = d_{01}^{-1} + d_{02}^{-1}$, $\sigma_3 = d_{01}^{-3} + d_{02}^{-3}$, and $\Delta_3 = d_{01}^{-3} - d_{02}^{-3}$.

When displacing the non-fixed vertex, the Hessian \mathbf{H}_U yields the rate at which the corrective forces increase when moving the vertex farther out of its rest pose \mathbf{x}_0 . Since we are interested in the worst case—the displacement that increases the forces the least—we have to consider the smallest eigenvalue λ_{\min} . The worst-case displacement (i.e., the displacement direction that the mechanism can resist the least) is then given by the corresponding eigenvector. We have

$$\lambda_{\min} = \frac{k_0}{2} \left(\sigma_1 - \sqrt{\sigma_1^2 + 4y_0^2 \left(L^3 \Delta_3^2 + 2Lx_0 \sigma_3 \Delta_3 + \sigma_3 \left(-\sigma_1 + \sigma_3 \|\mathbf{x}_0\|^2 \right) \right)} \right).$$

In order to ensure sufficient rigidity of the stable poses, we design the mechanism in such a manner that \mathbf{x}_0 lies at a position \mathbf{x}_0^* that maximizes λ_{\min} , i.e.,

$$\mathbf{x}_0^* = \arg \max_{\mathbf{x}_0} \lambda_{\min}(\mathbf{x}_0).$$

While the complex algebraic structure of λ_{\min} makes it difficult to obtain precise analytic solutions for \mathbf{x}_0^* , we can nevertheless obtain all relevant information by examining the function graph of $\lambda_{\min}(\mathbf{x}_0)$ as illustrated in Figure 8. The global optima with $\lambda_{\min} = \frac{\sqrt{2}}{L}$ are at the locations $(0, \pm \frac{L}{2})$, which corresponds to a right isosceles triangle. Further, we observe that high rigidity values can be found for values of \mathbf{x}_0 very close to a circle with radius $\frac{L}{2}$ centered at the origin. Such values correspond to the configurations that are very close to right triangles.

These insights influence the manner in which we construct our mechanisms in the following ways:

- (1) we penalize wedges resulting from Henneberg type I construction steps that are far from right angles, as they reduce the overall rigidity of stable poses;
- (2) we also penalize wedges resulting from Henneberg type I construction steps that are far from isosceles wedges, to prevent configurations that are close to right-angled but very flat, and

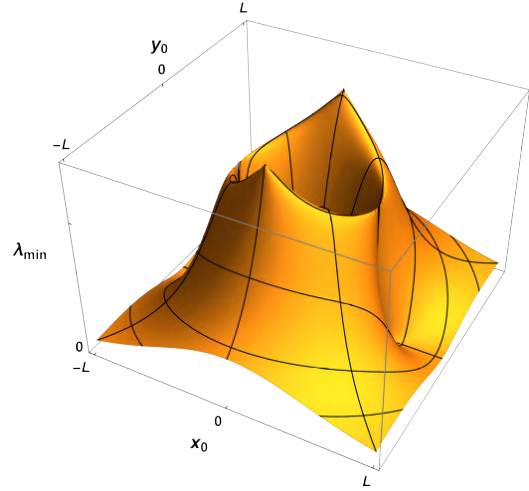


Fig. 8. Optimal rigidity of a wedge. The minimal eigenvalue λ_{\min} of the Hessian of the elastic energy is depicted for varying rest pose positions $\mathbf{x}_0 = (x_0, y_0)$ of the non-fixed vertex. The global maxima at $(0, \pm L/2)$ correspond to right isosceles wedges. The favorable rigidity of right wedges can be deduced from the high values on the circle with radius $L/2$.

- (3) we construct the core components, which orchestrate the multistable poses for multiple end effectors, with right isosceles wedges to guarantee maximal rigidity.

It must be noted that similar insights can be gained by an investigation of the rigidity matrix formalism for rigid graphs [Asimow and Roth 1978; Zelazo et al. 2012]. Since this field concerns itself exclusively with topological rigidity, these papers used a geometric distance function as the potential energy and construct the topological rigidity matrix based on it, then analyze the rank and eigenmodes of the rigidity matrix. Since they do not employ physically plausible quantities, we forgo such analysis here.

5 INVERSE STRUCTURE DESIGN

Equipped with the necessary tools from graph rigidity theory (see Section 4), in this section, we present our automatic system for multistable structure design. At the core of our method lies a mixed continuous/discrete optimization problem with the main objective of identifying a suitable minimally rigid framework (G, λ) that realizes the user-provided target poses. We denote the target pose sequence by $\mathbf{t}_{i,j}$, where we index end effector joints by $i = 1, \dots, N_E$ and the target poses by $j = 1, \dots, N_P$. For example, as illustrated in Figure 3 (a), the input SAILOR poses have $N_P = 2$ and $N_E = 4$ (i.e., two fixed and two non-fixed end effector joints). Apart from the joints, any bars of the end effectors will also form part of the framework; this is the case in Figure 3 (a), where each pair of end effector vertices are connected by an edge. For details, see Section 5.1.2.

In an abstract form, the optimization problem can be stated as

$$\begin{aligned} & \arg \min_{G, \lambda, \rho} E_{\text{tar}}(\rho, \mathbf{t}) \\ \text{s.t.} \quad & \left\{ \begin{array}{l} G \text{ is minimally rigid} \\ \lambda \text{ is consistent across all realizations } \rho \\ \text{transitions and rigidity are satisfactory} \\ \text{fabricability is ensured,} \end{array} \right. \quad (2) \end{aligned}$$

where E_{tar} measures the approximation error of the target positions, for which we employ an L^2 metric. Moreover, various constraints ensure that a working mechanism is obtained. First, as explained in Section 4.2, we require the underlying graph G of the mechanism to be minimally rigid, as this reduces the overall complexity of design while still providing rich multistable behavior. Further, as a physical bar-joint mechanism, each bar of the mechanism must retain its length across all stable poses; thus, the geometric realizations $\rho = \{\rho_1, \dots, \rho_{N_p}\}$ of the N_p poses must exhibit the same edge length labeling λ .

In order to ensure sufficient stability of each stable pose, a simplified version of the rigidity metric developed in Section 4.3 is employed. Moreover, we ensure feasible transitions between the stable poses by restricting our mechanism to configurations that do have a unique solution when using the forward symbolic simulation. Various fabrication requirements are considered and integrated as fabricability constraints into the optimization.

Section 5.1 describes our approach for identifying suitable minimally rigid graphs, which encompasses the discrete aspects of the optimization problem. All other constraints, as well as the objective function are formulated as continuous terms in Section 5.2. The overall solution strategy is summarized in Section 5.3.

5.1 Topology Construction

Different approaches might be used to identify suitable topologies of the mechanism's underlying minimally rigid graph. A general, albeit naïve, method would be to enumerate all Laman graphs that permit at least N_p realizations (see Section 4.2). From among these, one has to select a subset of N_p realization that are used for the individual poses; then select appropriate edges as compliant bars to enable the transitions between adjacent stable poses. This is already problematic since the number of realizations quickly grows with the vertex count; a general Laman graph with 13 vertices—less than the number of most examples we generated (see Table 1)—would exhibit in the order of 10^9 realizations. Moreover, one still has to assign special roles to certain vertices: (i) at least two vertices need to be *fixed*; (ii) N_E vertices must be marked as the *end effector* vertices; and (iii) a vertex must be selected as the *control* vertex by which a user manipulates the mechanism. It is easy to see that such an approach suffers from a combinatorial explosion in the number of possible candidate configurations that have to be checked, thereby making it intractable even for small N_p and N_E .

In contrast, we propose a construction-based method utilizing Henneberg rules for creating minimally rigid graphs (see Section 4.2). Further, we conceptually subdivide the mechanism into distinct components:

- A central multistable *switch*; this component provides at least N_p stable positions of a central *control vertex* and synchronizes the poses of all end effectors.
- A *coupler* for each end effector group; these components translate the positions of the switch's control vertex to the appropriate poses of the end effectors.

In contrast to the aforementioned naïve approach, this method is suited for our application scenarios of a moderate number of poses with rich articulation. While the comparatively small switch is designed to permit N_p stable states, the more complex couplers are fashioned to not add any more stable states to the mechanism, thereby avoiding undesired poses. Meanwhile, the decomposition into a central multistable switch and multiple couplers restricts the viable topologies to the combination of possible topologies of the switch and couplers. This also significantly prunes redundant graph topologies which are not suitable for synchronized multiple end effector structures. After describing the construction of the switch in Section 5.1.1 and of the couplers in Sections 5.1.2 and 5.1.3, we present the assembly of all components in Section 5.1.4.

5.1.1 Multistable Switch. In our switch-couplers scheme, the multistable switch is a topologically rigid and physically compliant bar-joint mechanism. It generates multiple stable positions for the central *control vertex* by switching the compliant bars. The central *control vertex* is shared with all couplers and actuates the movement of the end effectors through the couplers.

For the switch, we aim to identify a minimally rigid graph that realizes at least N_p distinct positions for the central *control vertex*. For a bistable switch, the solution is trivial: A three-node Laman graph (that is, a triangle) can realize two different central *control vertex* positions by flipping two edges relative to the third edge (which is fixed in the global coordinate system). Then, we extend this construction operation to an N_p -stable switch by rigidly fusing one of the flipping edges of a new bistable switch to the fixed vertices of an i -stable switch to form an $(i + 1)$ -stable switch. Appendix A provides a detailed algorithmic construction process for an arbitrary set of N_p positions of the control vertex.

As shown in Figure 2 (a), a physically compliant bar-joint multistable switch is obtained by transforming the bistable switches into triangles with two compliant bars; all the other edges are transformed to rigid bars. This enables a transition to all stable poses by sequentially switching the bistable switches.

5.1.2 End Effector Groups. As mentioned above, we integrate the rigid bodies of the end effectors as bars and joints into our mechanism. For this, we first collect connected end effector bodies into *end effector groups*. In the graph of the designed mechanism, these joints and bodies are then represented as vertices and edges with the same connectivity. Several possible end effector group variants are shown in Figure 9. Note that the user can also specify an end effector joint as fixed in space, which commonly happens in articulated objects such as those shown in Figures 12 to 14. By doing so, it would limit a connected rigid body to rotational motions only, as illustrated in Figure 9 (c).

5.1.3 Coupler Construction. Having specified the topological construction of the central multistable switch and the end effector

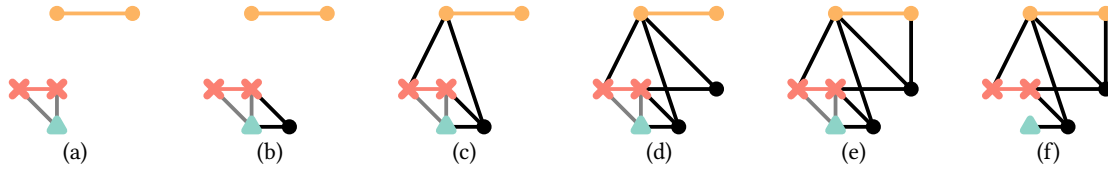


Fig. 10. Coupler topology construction. (a) Starting with a central control vertex (\blacktriangle) and two fixed vertices (\otimes), an initial triangle graph is defined. By applying the Henneberg type I construction rule (in this example twice, in (b) and (c)), a more complex Laman graph emerges. The added vertex in (c) is an end effector vertex (\bullet). (d) - (e) If the end effector group contains additional end effector joints, additional Henneberg type I steps are taken to generate the corresponding end effector vertices and edges. (f) Finally, the initial edges that connect the control vertex with the two fixed vertices are removed, resulting in a linkage. The control vertex is shared by all other couplers and the central multistable switch.

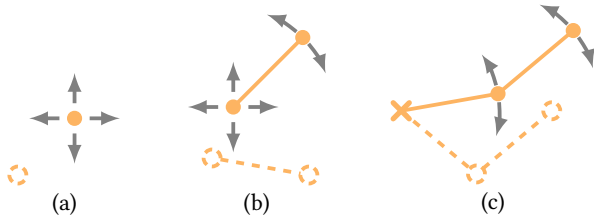


Fig. 9. End effector groups. Three exemplary end effector variants are shown, each of which would correspond to an *end effector group*. Spatial degrees of freedom are illustrated by arrows and a possible second pose is shown dashed. A fixed end effector vertex is shown in the end effector group (c), which is also utilized in the results shown in Figures 12 and 14.

groups in the last two sections, we still need to design suitable *coupler* graph topologies to translate the stable positions of the switch into the user-specified articulations of the end effectors. To this effect, we create a separate coupler for each end effector group, which all share the same *control vertex* of the multistable switch. An example of this can be seen in Figure 12, where the four end effector groups—one for each extremity—are connected to the central switch by four couplers.

As illustrated in Figure 10, we employ the following construction strategy: First, we specify the obligatory structures of each coupler: the *control vertex*, two *fixed vertices*, and the vertices and edges of the end effector group (see Figure 10 (a)). Here, the fixed vertices allow us to position the coupler in the global coordinate frame of the input model. From these structures, we ultimately design a linkage with two degrees of freedom that translates the movements of the control vertex into movements of the end effector (see Figure 10 (f)). However, we do not directly construct the linkage but utilize our framework of Laman graphs instead. Thus, we create as an intermediate result a minimally rigid coupler topology (see Figure 10 (e)) that is transformed into the corresponding linkage by removing the two edges connecting the fixed vertices and the control vertex. This approach has two advantages: (i) by Henneberg type I steps, we ensure symbolic computability of the kinematics (see Section 4.2); and (ii) we can use a coherent construction approach for all components of the multistable structure.

We want to recall that in this section, we concern ourselves with the construction of the *topology* of the couplers (i.e., the vertices and edges of the corresponding graph). The actual realization (i.e., the

positions of the vertices) of each stable pose will be found through continuous optimization later on as described in Section 5.2. Furthermore, we do not construct just a single coupler topology for each end effector group but we enumerate a set of possible topologies. Through our joint topological-geometrical scheme (see Section 5.3), we then identify promising topologies through optimization.

The construction of coupler topologies starts with the obligatory parts (see Figure 10 (a)) and proceeds by adding additional vertices through the application of Henneberg type I steps. By using increasing the number of steps, the complexity of the coupler topology is gradually increased. Furthermore, each step can connect the newly created vertex to different existing vertices. Keeping track of the already created vertices, this allows us to efficiently enumerate the possible coupler topologies for later optimization.

However, not all applications of Henneberg type I steps yield a sensible coupler topology. It has to be ensured that the end effector vertices and edges are part of the final graph (see Figure 10 (e)). This can be achieved by integrating subsequent end effector vertices together with the edge that connects them to the already integrated end effector vertices. An example is shown in Figure 10, where in the last construction step (see Figure 10 (e)), the second end effector vertex is added together with the edge that connects it to first end effector vertex that was integrated in a previous construction step (see Figure 10 (c)).

Between the integrations of end effector vertices and edges, arbitrary Henneberg type I steps can be applied (see Figure 10 (b,d)). Varying the nature and number of these steps yields the desired enumeration of possible coupler topologies. Two common sense rules should be applied here to avoid superfluous structures: After having integrated the last end effector vertex, the construction can be stopped. All subsequent vertices would not change the kinematic behavior of the end effector and unnecessarily complicate the coupler structure. At the same time, completely rigid subgraphs should be avoided (e.g., connecting a vertex to both fixed vertices) as such topologies do not provide any novel behavior that cannot be obtained with simpler structures.

As a minor detail, we also note that if the end effector group contains a vertex at a fixed position, it can substitute one of the fixed vertices. Examples for this are given in Figure 11 (a,b). We also found it beneficial to only use the control vertex in the first Henneberg type I step. This approach ensures that each coupler only has a single edge connecting to the control vertex. This avoids

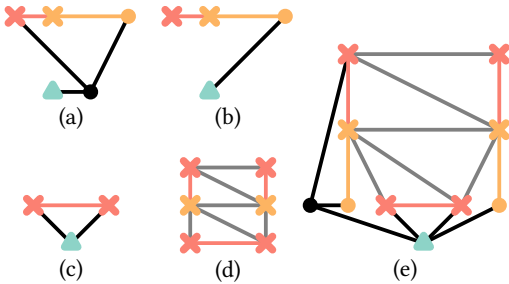


Fig. 11. Topology Assembly. The topology of the multistable mechanism is obtained by merging the coupler (a, b) and switch (c) graphs together with a minimally rigid graph that stabilizes all fixed vertices (d). In the final result (e), the control vertex (\blacktriangle) and the fixed vertices (\otimes) (as well as coinciding edges) of all components are merged. Fixed and non-fixed end effector nodes are represented by (\otimes) and (\bullet), while (\bullet) are standard vertices without a special role.

overconstraining the control vertex in the later optimization, which could occur with an increasing number of connecting edges.

5.1.4 Topology Assembly. Having generated a coupler topology for each end effector group, all their control vertices are merged into a single vertex, which we also merge with the control vertex of the multistable switch as shown in Figure 11. To complete the topology of the whole mechanism, we connect all fixed vertices of the couplers and the switch with a minimally rigid graph, which makes the overall structure again a Laman graph. Note that if the central multistable switch is replaced by a circular driving node, the assembled multistable mechanism becomes a circularly driven linkage. Thus, our method can also be used to discover feasible topologies for linkages without relying on elementary linkages as building blocks. It also allows us to leverage aspects of the optimization approaches from related works on linkage design. In the following section, we develop such a continuous constrained optimization problem within the context of multistable mechanisms.

5.2 Geometric Optimization

Having designed the topology of the mechanism as described in Section 5.1, one still needs to find suitable values for its geometric parameters, that is the placement of the vertices in the coordinate plane. We formulate this design task as a continuous optimization problem; in Section 5.2.1 the available degrees of freedom are described, which are optimized according to the objectives and constraints formulated in Section 5.2.2.

5.2.1 Geometric Parameters. There are different possibilities on how the mechanism can be parameterized: (i) according to edge lengths, where the parameters λ of an edge labeling are optimized together with three degrees of freedom for the overall position and orientation of the mechanism; or (ii) according to the actual positions of the vertices of the graph.

In our approach, we utilize the latter but sidestep the need for edge length-related constraints by utilizing the properties of our coupler topologies. The central multistable switch topology is determined by the required number of stable poses while the actual

edge lengths can be derived analytically from the required control vertex positions (see Section 5.1.1 and Appendix A).

Since the topology of the couplers is constructed using the Henneberg type I construction rule, their vertex movement can be simulated using symbolic kinematics (see Section 4.2). This enables us to analytically compute the positions of all coupler vertices by knowing the position of the fixed vertices and the control vertex as well as the lengths of all edges in the coupler topology.

Consequently, we only need to parameterize the vertex positions of a *rest pose* and vertex positions of *only* the control vertex in all other poses. In a pose different from the rest pose, the control vertex will assume a different position; together with the positions of all fixed vertices, we can utilize the vertex order and the edge lengths of the rest pose to analytically derive the positions of all other vertices of the mechanism. By construction, this ensures equal edge lengths in all poses.

5.2.2 Energies. With our topology construction strategy and chosen parameterization, we have already satisfied the first two constraints of our geometric optimization problem, as stated in Equation (2) (i.e., the underlying graph is minimally rigid, and the edge length labeling λ is consistent across all realizations). All remaining constraints are incorporated as soft constraints into a penalty method approach. We convert the constrained optimization problem into an unconstrained one of the form

$$\begin{aligned} & \arg \min_{\boldsymbol{\rho}} E(\boldsymbol{\rho}, \mathbf{t}) \\ E(\boldsymbol{\rho}, \mathbf{t}) = & E_{\text{tar}}(\boldsymbol{\rho}, \mathbf{t}) + \omega_{\text{tran}} E_{\text{tran}}(\boldsymbol{\rho}) + \\ & + \omega_{\text{rig}} E_{\text{rig}}(\boldsymbol{\rho}) + \omega_{\text{fab}} E_{\text{fab}}(\boldsymbol{\rho}), \end{aligned} \quad (3)$$

where E_{tar} measures the L^2 norm between the positions of the end effector vertices and their user-specified target position in each pose—that is,

$$E_{\text{tar}}(\boldsymbol{\rho}, \mathbf{t}) = \sum_j \|\boldsymbol{\rho}_j(v_i) - \mathbf{t}_{i,j}\|^2, \quad j = 1, \dots, N_P$$

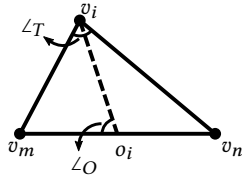
with i indexing all (non-fixed) end effector vertices.

The subdivision into couplers and a central multistable switch was motivated by the goal of avoiding undesired stable poses, of which a large number might exist in general minimally rigid frameworks. In order to ensure that only the stable states of the switch are present in the final mechanism, it must be guaranteed that couplers do not introduce additional stable states. In other words, the transition path of each coupler between two stable states must be unique. This is a known issue in linkage design [Thomaszewski et al. 2014], and a common approach is to prevent edges with common vertices from becoming collinear [Bächer et al. 2015]. Having built all coupler topologies with Henneberg type I steps, each of which adds an additional wedge to the existing graph, we define the desired transition energy E_{tran} based on these wedges. Subdividing the transition into N_S discrete steps, we add a penalization at each step s if a wedge's normalized area A (i.e., its area relative to its circumcircle area) falls below a threshold ϵ_{tran} . Thus, the transition energy term is given by

$$E_{\text{tran}} = \sum_s \sum_{i: A_i(\boldsymbol{\rho}_s) < \epsilon_{\text{tran}}} \left(\log \left(\frac{A_i(\boldsymbol{\rho}_s)}{\epsilon_{\text{tran}}} \right) \right)^2,$$

where $A_i(\rho_s)$ denotes the area of the i -th wedge, given its realization in transition step s . The logarithmic function here ensure a steep barrier near 0; a similar formulation was also used in [Bächer et al. 2015].

As derived in Section 4.3, an isosceles right wedge has the highest rigidity. Thus, we penalize wedges that deviate too far from this guideline using the rigidity energy term E_{rig} . For each wedge, as illustrated on the right, which originated from the application of a Henneberg type I step by adding a vertex v_i to the mechanism and connecting to existing vertices v_m and v_n , we denote with $\angle T_i$ the interior angle of the triangle at vertex v_i and $\angle O_i$ as the angle spanned from the middle point of v_m and v_n to v_i .



Similar to E_{tran} above, we consider values of $\cos(\angle)$ above the threshold ϵ_{rig} as undesirable and define

$$E_{\text{rig}} = \sum_j \left(\sum_{i: |\cos(\angle T_i(\rho_j))| > \epsilon_{\text{rig}}^T} \left(\log \left(\frac{|\cos(\angle T_i(\rho_j))|}{\epsilon_{\text{rig}}^T} \right) \right)^2 + \sum_{i: |\cos(\angle O_i(\rho_j))| > \epsilon_{\text{rig}}^O} \left(\log \left(\frac{|\cos(\angle O_i(\rho_j))|}{\epsilon_{\text{rig}}^O} \right) \right)^2 \right),$$

where ρ_j denotes the realization at each stable pose j and i indexes all wedges.

To ensure fabricability and ease of use of our mechanism, we enforce box constraints $\epsilon_{\text{length},\min}$ and $\epsilon_{\text{length},\max}$ on the edge lengths of the mechanism and penalize control vertex positions outside a rectangular bounding region given by $[\epsilon_{x,\min}, \epsilon_{x,\max}] \times [\epsilon_{y,\min}, \epsilon_{y,\max}]$. Any value outside these intervals adds a quadratic contribution to the overall fabricability energy E_{fab} —that is, a term $(x - \epsilon)^2$ is added if $x < \epsilon$ or $x > \epsilon$ (depending on the constraint).

5.2.3 Initialization. Given the topology of a mechanism as input to our continuous optimization problem, we still have to choose a set of suitable initializations of ρ to explore the different realizations of the topology. With the aforementioned addition of the constraint E_{tran} , we ensure unique transition paths between stable poses. For minimally rigid graphs built with Henneberg type I steps, this is equivalent to the requirement that none of the wedges, which are added by these steps, change their orientation. As a consequence, our continuous solver will only converge to a solution that exhibits the same wedge orientations as the initialization. For example, only one of the realization as shown in Figure 6 would be accessible as all others require the flip of one or more wedges.

To ensure that our solution space covers all possible realizations, we initialize our continuous optimization problem with all wedge orientations. Recalling the coupler construction in Section 5.1.3 (see Figure 10), we need—apart from the initial triangle graph— n additional Henneberg type I steps to construct the coupler topology. To initialize the positions of the initial triangle, we choose the center of the user-specified model as the initial position of the control vertex, and sample random locations in the vicinity as initial positions for the fixed vertices.

Additionally, motivated by the derivation in Section 4.3, the initial shape of the wedge is restricted to an right isosceles wedge to ensure the maximal rigidity of each stable configuration. As initial configuration for each of the n Henneberg type I steps in the construction of both central switch and couplers, we construct a corresponding right isosceles wedge. After choosing an orientation for each wedge, this analytic construction yields a fixed position for all the vertex positions of these triangles. Together with the choice of control and fixed vertex positions, this gives the initialization value of ρ in Equation (3). Thus, ρ is initialized with optimal rigidity (i.e., smallest E_{rig}) but with end effector vertices not matching their target positions. Given the small number n of required Henneberg type I steps in our context—a consequence of our subdivision of the mechanism into various components—we can exhaustively check all 2^n possible wedge orientations as initializations.

5.2.4 Implementation. We solve the optimization problem Equation (3) using a standard BFGS solver with automatic differentiation as implemented in Ceres Solver [Agarwal et al. 2019]. We treat E_{tran} and E_{fab} as hard constraints and double their weights ω_{tran} and ω_{fab} in consecutive runs up to a maximum limit, as long as the respective constraints are not satisfied. E_{rig} is considered a soft constraint with fixed weights ω_{rig} . The derivatives of all energies can be computed symbolically; one reason being the established vertex order when constructing the topologies. Due to the non-convexity of the problem, we sample up to 16 different initial configurations for the fixed vertices positions.

In practice, we set $\omega_{\text{tran}} = 100$, $\omega_{\text{fab}} = 10$, and $\omega_{\text{rig}} = 10$. $\epsilon_{\text{length},\min}$ is set to 12.5 mm, while $\epsilon_{\text{length},\max}$ as well as the rectangular bounding region depend on the region that is covered by the user-specified end effector movement. The threshold ϵ_{tran} is set to 0.01, ϵ_{rig}^O and ϵ_{rig}^T are set to $\cos 75^\circ$, and the number of discrete steps N_S is set to 5 between every adjacent stable poses. Due to the non-convex nature of the optimization problem, there could be extreme situations where the entire energy converged but the hard constraint parts remain non-zero. Our system informs the user when the hard constraint terms do not converge to zero.

5.3 Joint Topological-Geometrical Optimization

As presented in Section 5.1.3, the coupler design requires the selection of a minimally rigid graph in each construction operation. After making such selections, the resulting topology of the mechanism is subject to a continuous optimization problem to identify optimal positions for the joints of the mechanism. We observed that the achievable quality of the result of the continuous optimization is highly dependent on the chosen topology. However, there is no simple heuristic that guarantees feasible topologies for all application cases. Instead, we solve this problem in an alternating energy descent fashion, i.e. iteratively optimize for the graph topology selection and the continuous parameters by turns.

We perform the aforementioned graph selection among a set \mathcal{T} of suitable minimally rigid graphs. Since we require that couplers are generated by the Henneberg type I construction rule, we precisely select type I Laman graphs as elements of \mathcal{T} . For improved

computational efficiency of the selection, small sizes of \mathcal{T} are preferred. For our purposes—and all our examples—we first filter out the redundant topologies, such as graphs with unused sub-graphs (this is, subgraphs that do not influence the end effector vertex positions) or mismatch with the end effector connectivity. We then select a reasonable number of candidates \mathcal{T} . We found it sufficient to choose as elements for \mathcal{T} all type I Laman graphs with four and five vertices, as well as five random ones with six vertices.

Improving this topology selection constitutes the outermost loop of our design method. Given $N_{E,N}$ non-fixed end effector vertices by the user, we initialize the parameter vector τ with a random choice of topologies from $\mathcal{T}^{N_{E,N}}$. In each iteration of the outer loop, we replace a random element of τ with the optimal choice from \mathcal{T} . When replacing, for example, the i -th element τ_i , we iterate through all possible replacements $\tau_j \in \mathcal{T} \setminus \{\tau_i\}$ and perform topology construction and geometric optimization, according to Sections 5.1 and 5.2. We then replace τ_i with the new topology τ_j that yielded the smallest residual energy when solving Equation (3).

This iteration is continued until one of the common stopping criteria is reached: (i) the residual energy falls below a specified threshold; (ii) no replacement yields an improvement; or (iii) the maximal iteration count is reached.

As a result of our topological and geometric optimization, we obtain a graph of the mechanism together with a geometric embedding of all its vertices in each pose.

6 FABRICATION

Before fabricating a physical realization of the optimized bar-joint mechanism, additional post-processing steps are necessary. So far, collisions among the various parts during the transitions between stable poses have not been considered. We resolve this issue by placing bars in different layers, an approach commonly utilized in linkage design [Coros et al. 2013]. As detailed in Section 6.1, we detect collisions between the various parts of the mechanism and assign conflicting parts to different layers. Subsequently, we generate actual 3D geometry for each part of the mechanism and add them to the user-specified model. From these, we derive build instructions for the fabrication method of choice (e.g., 3D printing or laser cutting). After manufacturing, the final models can be assembled and immediately used.

6.1 Collision Avoidance

In order to detect potential collisions between the bars and joints of the mechanism, we convert the edges and vertices of the mechanism’s graph into two-dimensional proxy shapes. We use a simple safety margin of length r around each element, where we select r as the radius of the final geometry (see Section 6.2) and add a small safety margin of 0.125 mm. Conceptually, this corresponds to taking the Minkowski sum of each edge and vertex with a disk of radius r . Leveraging the symbolic computability of our mechanism, we simulate each transition and check if any two proxy shapes overlap; in this case, we mark them as overlapping.

Similar to the work of Bächer et al. [2013], we then formulate a constraint satisfaction problem that assigns a layer index to each bar, while ensuring that any colliding elements are placed in different

layers. Additionally, we ensure that the end effector vertices and bars can be connected to the end effector parts of the user-provided model, which is placed in the topmost layer. As an overall objective, we aim for the smallest number of layers that permits a solution to this problem, thereby reducing the overall fabrication complexity and the ‘thickness’ of the mechanism. We apply integer programming to this problem and utilize Google’s OR-Tools [Perron and Furnon 2019].

For fabrication convenience and increased usability, we separate the multistable switch from the remainder of the mechanism and explicitly place it in the bottom-most layers. This makes the switch easily accessible for actuation, since it can be reached from the back of the model. Further, it can be fabricated separately and easily assembled, since it connects to the remainder of the mechanism through a single joint—the control vertex. While this separation increases the number of required layers, this is generally not a prominent effect, since the multistable switch usually overlaps considerable parts of the coupler region and a large number of collisions have to be avoided anyhow.

Finally, we consider the supporting structures that hold fixed vertices in place. We use a simple heuristic for this problem by iterating possible support structures and perform the aforementioned collision detection and layering in each iteration. Once we identify feasible support structures for all fixed nodes, we terminate the search.

6.2 Model Generation

Having identified a suitable layering of the mechanism, we convert the edges and vertices to 3D geometry and fabrication blueprints. For 3D printing, each edge is converted into a rounded bar with a radius of 3 mm and each vertex into a capped axle that spans all the layers between the bars that it connects as a hinge joint. We add an inter-layer clearance of 0.25 mm to avoid unintended contact between parts of adjacent layers. Optionally, bars of rigid subgraphs (e.g., a triangle subgraphs) are fused together for increased stability. In the multistable switch, we utilize curved bars as compliant elements. Adequate tolerances are selected empirically; both the switch and the remainder of the mechanism can be fabricated in a single piece.

For laser cutting, we use a combination of metal rods and ball bearings as hinge joints. Thus, we only output the projected shape of the rounded bars with a radius of 6 mm as a fabrication blueprint, and all bars are assigned a layer thickness of 3 mm.

7 RESULTS

7.1 Designs

We applied our method to a variety of figurine examples. In our tests, for the FLAG SEMAPHORE ‘SIG’ the optimization converged with the residual energy falling below the specified threshold. In all remaining examples, our algorithm stopped when no replacement topology yielded an improvement. General statistics related to the results can be found in Table 1. All 3d printed physical realizations were fabricated with a Stratasys J750 full-color 3D printer using the materials of the Stratasys Vero family. The laser cutted results were made of 4 mm POM material cut with an Epilog Fusion Pro laser machine.

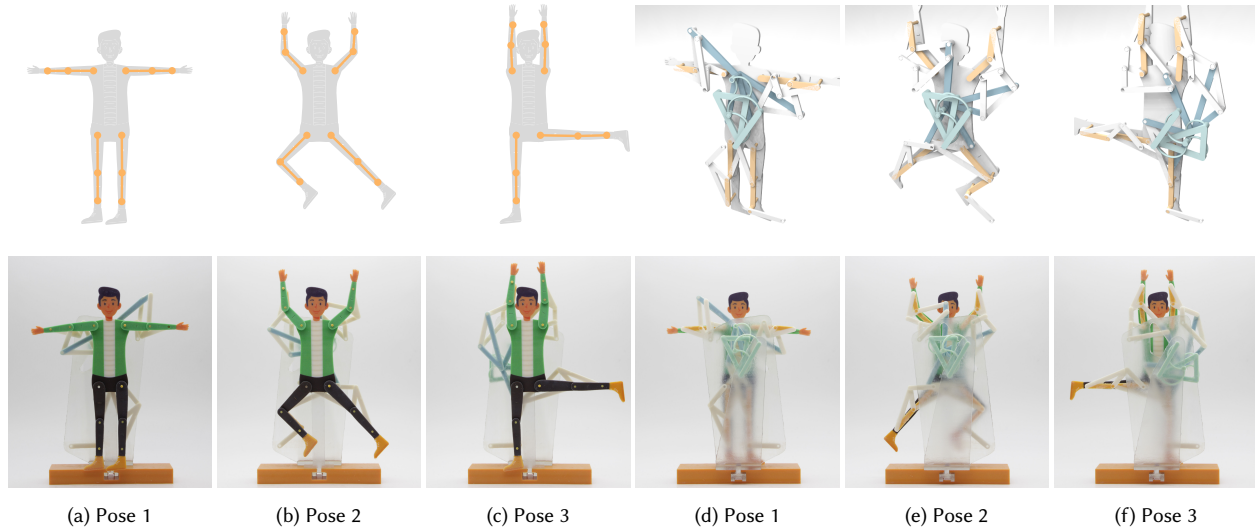


Fig. 12. FIGURINE result (top row: rendered; bottom row: physical artifact). The three poses are shown from the front and the back.

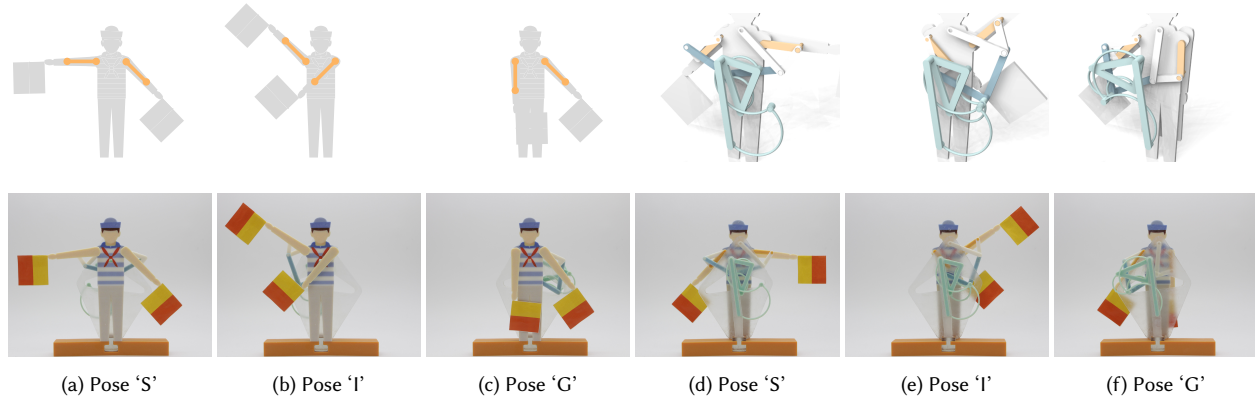


Fig. 13. FLAG semaphore result (top row: left: input poses (gray) and generated end effector positions in generated graph realizations, right: rendered compliant joint-bar mechanism; bottom row: physical artifact). The three poses are shown from the front and the back.

Table 1. Results statistics. We provide the geometric complexity of both in- and output as vertex and edge counts ($\#V$ and $\#E$) as well as the target approximation and rigidity energies E_{tar} and E_{rig} of each result. The maximal deviation of any end effector joint from its user-specified target position is presented with Δ_{max} . The last column reports the runtime of our design method.

Result	Poses	End effector		Coupler		Switch		E_{tar}	E_{rig}	Δ_{max} [mm]	Time
		$\#V$ (fixed)	$\#E$	$\#V$	$\#E$	$\#V$	$\#E$				
FLAG SIG	3	4 (2)	2	8	6	6	9	0.36	5.62	0.09	4 s
FIGURINE	3	12 (4)	8	24	30	6	9	177.01	17.37	1.54	18 min
ICONIC	3	12 (4)	8	26	34	6	9	607.90	20.58	2.86	21 min
GRIPPER	3	6 (2)	4	15	20	6	9	240.83	0.11	1.76	1 min

In Figures 1 and 14, three ICONIC poses of famous athletes were transferred to a human model with movable limbs. Each extremity forms an end effector group consisting of two edges and three vertices, one of which is kept fixed at the shoulders and hip (see

Figure 9 (c)). Our automatic algorithm identifies a working mechanism that successfully transitions between the poses and exhibits a potential energy minimum at each of them. The largest deviation of an end effector vertex from its target position was 2.86 mm, which

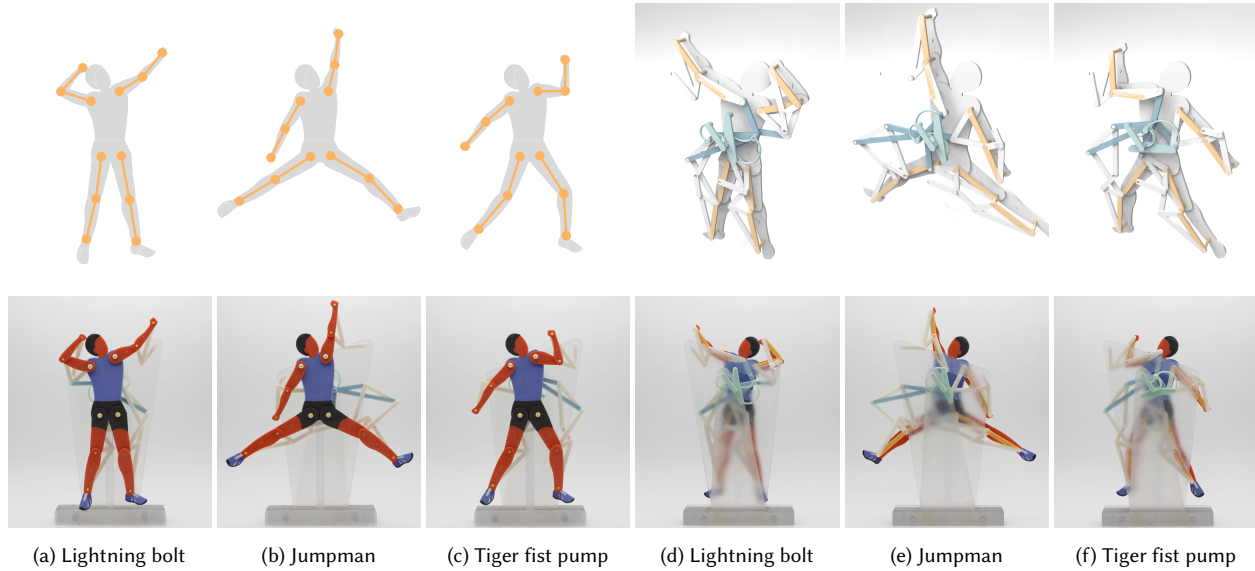


Fig. 14. ICONIC poses result (top row: rendered; bottom row: physical artifact). The three poses are shown from the front and the back.



Fig. 15. GRIPPER poses result (top row: rendered; bottom row: physical artifact). The three poses are shown from the front and the back.

corresponds to $\approx 1\%$ of the height of the model, thereby indicating excellent agreement.

A similar setup was used for the FIGURINE in Figure 12. Here, we show that our mechanism is able to replicate a combination of different end effector movement patterns: (i) positions that stay identical in two poses but vary in another (right leg); (ii) symmetric movement of two end effector groups (arms); and (iii) independent movement of other end effector groups (left leg). Again, excellent matching is achieved with a deviation of less than 1.6 mm.

A model with fewer end effector groups was used to reproduce FLAG semaphore signaling. Each arm is represented by an end effector group with two vertices and one edge. We took the three-letter combination ‘SIG’ and translated it into the corresponding three flag poses, as illustrated in Figure 13. Given the reduced complexity,

our algorithm was able to find a perfect solution with virtually no deviation in less than 5 s.

As a prototype of applying multistable compliant structure to functional mechanisms, as shown in Figure 15, a tri-stable mechanical gripper model was designed with three distinct poses, Standby, Grasp and Release. A feasible structure with a maximum deviation of 1.76 mm was generated, and proved to be capable of stably holding a 100g massage ball in our experiment (please see the accompanied demo video).

7.2 Validations

In order to investigate the overall behavior of our design algorithm, we applied it to a large number of inputs. Extending the FLAG semaphore example, which we already showed for the characters ‘SIG’ in Figure 13, we applied our method to *all* character

sequences of three alphanumeric characters or less. From among these 47 988 sequences, we identified 9152 unique ones by omitting symmetric and identical sequences (as various characters map to the same flag semaphore). Our method successfully generated a working mechanism for all of these. In the supplemental material, a gallery of these results can be found.

The average runtime per example was below 9 s on a regular workstation with an quad-core Intel Xeon E3-1230 v5 and 32 GB RAM. The worst-case deviation of an end effector joint from its target position was 6.04 mm (for a model size of ≈ 150 mm), which documents excellent agreement. For 99% of the flag semaphores in the gallery, the distances between input and optimized end effector positions are smaller than 1 mm, corresponding to less than 0.7% of the scale of the input model. To quantify the stability of the stable poses, we performed physical measurements of the required forces for local perturbations by hanging weights on joints within the fabricated models. We observed that the response force for unit length(mm) perturbations ranges from 0.1612N to 1.2135N for the laser cut models (measured on one of the end effectors and the central switch node of GRIPPER), for the 3D printed models (measured on one of the end effectors and the central switch node of FLAG SEMAPHORE), it ranges from 0.1902N to 0.3582N. We also tested our algorithm on more challenging inputs, including more stable poses, more end effectors, and asymmetric scales of different end effectors. Due to the manufacturing limitations for such complicated results, we only provide virtual results. The additional test are presented in the supplemental material.

8 DISCUSSION

In this paper, we introduced a novel method for designing planar multistable compliant structures. Given a number of desired stable states and a set of positions for each state as input, our goal was to identify a topology and geometric realization of a compliant mechanism consisting of bars and joints that has an energy minimum at each corresponding state and enables transformations between the positions by passing a finite energy barrier.

Our technique is based on insights from minimally rigid graph theory, which enables us to identify effective topologies and thereby obtain a tractable design space which further guarantees that the kinematics of all designs can be evaluated symbolically.

Regarding the complexity of our algorithm, on the one hand, the graph selection part is a combinatorial problem, which means that the computational time is exponential with respect to the total number of nodes. On the other hand, the complexity of continuous optimization part is linear with respect to the number of stable poses. Therefore, the overall complexity of our algorithm is exponential in the number of input end effectors and close to linear in the number of stable poses. In practice, the computational time is also slightly influenced by the random initial configurations.

Further, we propose a gradient-based optimization scheme for identifying a realization of the topology by utilizing an easy-to-evaluate approximation of the rigidity of the stable poses.

We successfully applied our method for designing physical figurines that can transform between a discrete set of stable poses with anticipated order. The actuation is usually triggered by moving a

node of the switch, which then propagates the motion to the end effectors. This design choice as well as the actual construction of switch and couplers are motivated by a balance of topological rigidity, physical rigidity and computational complexity. Prioritizing a minimal number of bars and joints over all these other concerns might lead to challenging combinatorial optimization problems for which it is unclear if computationally tractable solutions do exist. An interesting avenue for future research would be to study the inverse actuation chain—that is, to move the mechanism by actuating one of the end effectors.

Currently, we do not incorporate structural analysis to ensure the robustness but rely on empirical experience for setting parameters such as the thickness of bars. Apart from optimizing for structural robustness, in the future it would be interesting to also tune the deformation behavior of elastic bars by adjusting their shape, thereby obtaining additional degrees of freedom for tuning the energy landscape of the mechanism.

Our system performs layering of potentially colliding elements in a post-process. Although the layerings of all the presented examples are automatically computed, there is no guarantee for the existence of a valid layering for an arbitrary given mechanism. Currently such problems require to be fixed manually by, for example, changing the geometry of rigid bars. We believe that this could also be automated. However, an interesting extension of our approach would be to provide high-level artistic control over the placement of individual elements.

Although our algorithm generates theoretically sound results, fabrication can still be challenging. For example, in PolyJet 3D printing, it is difficult to perfectly remove all support material within joints. Furthermore, required tolerances to prevent parts from fusing during 3D printing, small inaccuracies of laser-cut parts, and bending can result in undesired out-of-plane motion.

Finally, while not trivial, an exciting research topic would be to generalize the system to 3D mechanisms. Despite its limitations, we believe our method is an essential contribution to paving the road toward multistability design for a broad spectrum of applications. For example, we plan to use our system for designing multifunctional materials—that is, materials that can vary and reconfigure their properties, deployable structures, and general shape transformers.

ACKNOWLEDGMENTS

We would like to thank everyone who contributed to this paper, the authors of artworks for all the examples, including @macrovector_official and Wikimedia for the FLAG semaphore, and @pikisuperstar for the FIGURINE. The photos of iconic poses in the teaser were supplied by (from left to right): Mike Hewitt/Olympics Day 8 - Athletics/Getty Images, Oneinchpunch/Basketball player training on a court in New York City/Shutterstock, and Andrew Redington/Tiger Woods/Getty Images. We also want to express our gratitude to Christian Hafner for insightful discussions, the IST Austria machine shop SSU, all proof-readers, and anonymous reviewers. This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie grant agreement No 642841 (DISTRO), and under the European Research Council grant agreement No 715767 (MATERIALIZABLE).

REFERENCES

- Sameer Agarwal, Keir Mierle, and Others. 2019. Ceres Solver. Online. <http://ceres-solver.org>
- L. Asimow and B. Roth. 1978. The rigidity of graphs. *Trans. Amer. Math. Soc.* 245 (1978), 279–289. <https://doi.org/10.1090/s0002-9947-1978-0511410-9>
- Moritz Bäcker, Bernd Bickel, Doug L. James, and Hanspeter Pfister. 2012. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph.* 31, 4 (jul 2012), 1–9. <https://doi.org/10.1145/2185520.2185543>
- Moritz Bäcker, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: interactive linkage editing using symbolic kinematics. *ACM Trans. Graph.* 34, 4 (July 2015), 99:1–99:8. <https://doi.org/10.1145/2766985>
- Nakul Prabhakar Bende, Arthur A. Evans, Sarah Innes-Gold, Luis A. Marin, Itai Cohen, Ryan C. Hayward, and Christian D. Santangelo. 2015. Geometrically controlled snapping transitions in shells with curved creases. *Proc. National Academy of Sciences* 112, 36 (Sept. 2015), 11175–11180. <https://doi.org/10.1073/pnas.1509228112>
- Amit H. Bermanno, Thomas Funkhouser, and Szymon Rusinkiewicz. 2017. State of the Art in Methods and Representations for Fabrication-Aware Design. *Computer Graphics Forum* 36, 2 (may 2017), 509–535. <https://doi.org/10.1111/cgf.13146>
- Gaurav Bharaj, Danny M. Kaufman, Etienne Vouga, and Hanspeter Pfister. 2018. Metamorphs: Bistable Planar Structures. *CoRR abs/1804.06996* (2018). arXiv:1804.06996 <http://arxiv.org/abs/1804.06996>
- Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. 2018. State of the Art on Stylized Fabrication. *Computer Graphics Forum* 37, 6 (feb 2018), 325–342. <https://doi.org/10.1111/cgf.13327>
- Jose Capco, Matteo Gallet, Georg Grasegger, Christoph Koutschan, Niels Lubbes, and Josef Schicho. 2018. The number of realizations of a Laman graph. *SIAM Journal on Applied Algebra and Geometry* 2, 1 (Jan. 2018), 94–125. <https://doi.org/10.1137/17M1118312> arXiv: 1701.05500.
- Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph.* 32, 6 (nov 2013), 1–11. <https://doi.org/10.1145/2508363.2508400>
- Guimin Chen, Yanjie Gou, and Aimei Zhang. 2011. Synthesis of compliant multistable mechanisms through use of a single bistable mechanism. *Journal of Mechanical Design* 133, 8 (2011), 081007. <https://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1450639>
- Tian Chen, Jochen Mueller, and Kristina Shea. 2017. Integrated Design and Simulation of Tunable, Multi-State Structures Fabricated Monolithically with Multi-Material 3D Printing. *Scientific Reports* 7, 1 (mar 2017). <https://doi.org/10.1038/srep45671>
- Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. *ACM Trans. Graph.* 32, 4 (July 2013), 1. <https://doi.org/10.1145/2461912.2461953>
- Jérémie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, and Christian Dick. 2015. By-example synthesis of structurally sound patterns. *ACM Trans. Graph.* 34, 4 (jul 2015), 137:1–137:12. <https://doi.org/10.1145/2766984>
- Mohammad R. Golabchi and Simon D. Guest. 2009. Morphing Multistable Textured Shells. In *Proc. International Association for Shell and Spatial Structures Symposium (IASS)*. 1–12.
- Lebrecht Henneberg. 1908. Die Graphische Statik der Starren Körper. In *Encyklopädie der Mathematischen Wissenschaften mit Einschluss ihrer Anwendungen*, Felix Klein and Conrad Müller (Eds.). Mechanik, Vol. 4. Springer Fachmedien Wiesbaden GmbH, Wiesbaden, Chapter 5, 345–434. https://doi.org/10.1007/978-3-663-16021-2_5
- C. Huber, C. Abert, F. Bruckner, M. Groenefeld, O. Muthsam, S. Schuschnigg, K. Sirak, R. Thanhoffer, I. Teliban, C. Vogler, R. Windl, and D. Suess. 2016. 3D print of polymer bonded rare-earth magnets, and 3D magnetic field scanning with an end-user 3D printer. *Applied Physics Letters* 109, 16 (oct 2016), 162401. <https://doi.org/10.1063/1.4964856>
- Agustin Iniguez-Rabago, Yun Li, and Johannes T. B. Overvelde. 2019. Exploring multistability in prismatic metamaterials through local actuation. *Nature Communications* 10, 1 (dec 2019). <https://doi.org/10.1038/s41467-019-13319-7>
- Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. 2017. Digital Mechanical Metamaterials. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, Denver, Colorado, USA, 977–988. <https://doi.org/10.1145/3025453.3025624>
- B. D. Jensen, L. L. Howell, and L. G. Salmon. 1999. Design of two-link, in-plane, bistable compliant micro-mechanisms. *Journal of Mechanical Design* 121, 3 (1999), 416–423. <https://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleID=1445668>
- Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid deployment of curved surfaces via programmable auxetics. *ACM Trans. Graph.* 37, 4 (jul 2018), 1–13. <https://doi.org/10.1145/3197517.3201373>
- Gerard Laman. 1970. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics* 4, 4 (oct 1970), 331–340. <https://doi.org/10.1007/bf01534980>
- Seubpong Leelavanichkul, Andrej Cherkaev, Daniel O. Adams, and Florian Solzbacher. 2010. Energy absorption of a helicoidal bistable structure. *Journal of Mechanics of Materials and Structures* 5, 2 (aug 2010), 305–321. <https://doi.org/10.2140/jomms.2010.5.305>
- Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-band topology optimization on a sparsely populated grid. *ACM Trans. Graph.* 37, 6 (dec 2018), 1–14. <https://doi.org/10.1145/3272127.3275012>
- Thomas Lumpe and Kristina Shea. 2019. Computational Design of 4D Printed Shape Morphing Multi-State Lattice Structures. In *Volume 2A: 45th Design Automation Conference*. American Society of Mechanical Engineers. <https://doi.org/10.1115/detc2019-97774>
- Li-Ke Ma, Yizhong Zhang, Yang Liu, Kun Zhou, and Xin Tong. 2017. Computational design and fabrication of soft pneumatic objects with desired deformations. *ACM Trans. Graph.* 36, 6 (nov 2017), 1–12. <https://doi.org/10.1145/3130800.3130850>
- Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural voronoi foams for additive manufacturing. *ACM Trans. Graph.* 35, 4 (jul 2016), 1–12. <https://doi.org/10.1145/2897824.2925922>
- Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. 2014. ChaCra: An Interactive Design System for Rapid Character Crafting. <https://doi.org/10.2312/sca.20141130>
- Vittorio Megaro, Jonas Zehnder, Moritz Bäcker, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. 2017. A computational design tool for compliant mechanisms. *ACM Trans. Graph.* 36, 4 (jul 2017), 1–12. <https://doi.org/10.1145/3072959.3073636>
- Gen Nishida, Adrien Bousseau, and Daniel G. Aliaga. 2019. Multi-Pose Interactive Linkage Design. *Computer Graphics Forum* 38, 2 (2019). <https://doi.org/10.1111/cgf.13637>
- Zherong Pan, Min Liu, Xifeng Gao, Kai Xu, and Dinesh Manocha. 2019. Globally Optimal Joint Search of Topology and Trajectory for Planar Linkages. *arXiv:1905.08956 [cs]* (May 2019). <http://arxiv.org/abs/1905.08956> arXiv: 1905.08956.
- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph.* 34, 4 (jul 2015), 135:1–135:12. <https://doi.org/10.1145/2766937>
- Laurent Perron and Vincent Furnon. 2019. OR-Tools. Online. <https://developers.google.com/optimization/>
- Hilda Pollaczek-Geiringer. 1927. Über die Gliederung ebener Fachwerke. *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik* 7, 1 (1927), 58–72. <https://doi.org/10.1002/zamm.19270070107>
- Ahmad Rafsanjani and Damiano Pasini. 2016. Bistable auxetic mechanical metamaterials inspired by ancient geometric motifs. *Extreme Mechanics Letters* 9 (dec 2016), 291–296. <https://doi.org/10.1016/j.eml.2016.09.001>
- T. Schioler and S. Pellegrino. 2007. Space Frames with Multiple Stable Configurations. *AIAA Journal* 45, 7 (jul 2007), 1740–1747. <https://doi.org/10.2514/1.16825>
- Yuanping Song, Robert M. Panas, Samira Chizari, Lucas A. Shaw, Julie A. Jackson, Jonathan B. Hopkins, and Andrew J. Pascall. 2019. Additively manufacturable micro-mechanical logic gates. *Nature Communications* 10, 1 (Feb. 2019), 882. <https://doi.org/10.1038/s41467-019-08678-0>
- Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational design of linkage-based characters. *ACM Trans. Graph.* 33, 4 (July 2014), 1–9. <https://doi.org/10.1145/2601097.2601143>
- Andreas Wingert, Matthew D. Lichter, Steven Dubowsky, and Moustapha Hafez. 2002. Hyper-redundant robot manipulators actuated by optimized binary-dielectric polymers. In *Smart Structures and Materials 2002: Electroactive Polymer Actuators and Devices (EAPAD)*, Yoseph Bar-Cohen (Ed.). SPIE. <https://doi.org/10.1117/12.475189>
- Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bäcker. 2018. Bend-it: design and fabrication of kinetic wire characters. *ACM Trans. Graph.* 37, 6 (dec 2018), 1–15. <https://doi.org/10.1145/3272127.3275089>
- Daniel Zelazo, Antonio Franchi, Frank Allgöwer, Heinrich Bühlhoff, and Paolo Robuffo Giordano. 2012. Rigidity Maintenance Control for Multi-Robot Systems. In *Proceedings of Robotics: Science and Systems*, Nicholas Roy, Paul Newman, and Siddhartha Srinivasa (Eds.), Vol. VIII. MIT Press, Sydney, Australia, 1–8. <https://doi.org/10.15607/RSS.2012.VIII.060>
- Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel. 2017. Functionality-aware retargeting of mechanisms to 3D shapes. *ACM Trans. Graph.* 36, 4 (jul 2017), 1–13. <https://doi.org/10.1145/3072959.3073710>
- Bo Zhu, Méline Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 5 (jul 2017), 1–16. <https://doi.org/10.1145/3095815>

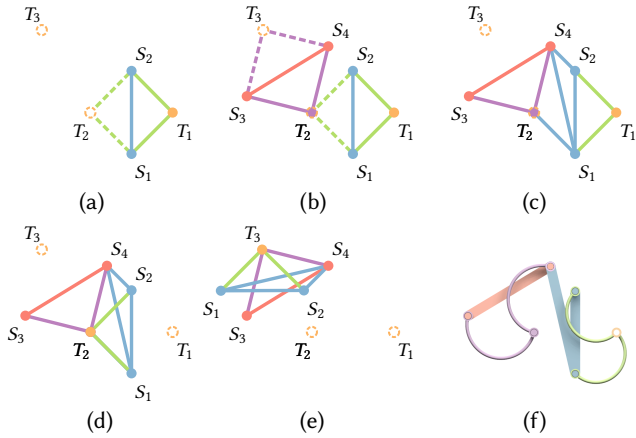


Fig. 16. Multistable switch construction. (a) A flippable isosceles right triangle is constructed as the bistable switch for targets T_1 and T_2 . (b) The second bistable switch is constructed for targets T_2 and T_3 . (c) Two bistable switches are merged into a multistable switch. (d) and (e) The other two stable configurations of the multistable switch. (f) A compliant bar-joint mechanism is generated based on the multistable switch graph (the edges T_2S_1, S_2S_4 are omitted in the physical construction by fusing the rigid bars S_1S_2, S_1S_4 and compliant bar T_2S_4 at S_1 and S_4).

A MULTISTABLE SWITCH CONSTRUCTION

Given a sequence of N_T target positions $\mathbf{t}_1, \dots, \mathbf{t}_{N_T}$, we present an algorithm that generates a rigid framework (V, E) , which realizes these positions with its end effector vertex v_e . It generates a design where only a single element in the framework is flipped when transitioning v_e between consecutive target locations. We exclude degenerated input by assuming that consecutive target positions are sufficiently far apart, i.e., $\|\mathbf{t}_{i-1} - \mathbf{t}_i\| > \varepsilon_T$.

A.1 Bistable Switch

Starting with the simplest case of only two targets positions \mathbf{t}_1 and \mathbf{t}_2 , as shown in Figure 16 (a), we construct a framework consisting of only a single isosceles right triangle. For this, we denote the vertices for \mathbf{t}_1 and \mathbf{t}_2 as T_1 and T_2 . We define a square with edge T_1T_2 as one of its diagonals. The other diagonal of the square is denoted as S_1S_2 , and its two end points constitute the first two vertices of the switch. Using these, we construct our first bistable switch $\Delta S_1S_2T_1$. The transition between the two target positions \mathbf{t}_1 and \mathbf{t}_2 can be achieved by fixing the edge S_1S_2 , thereby flipping the triangle $\Delta S_1S_2T_1$ to $\Delta S_1S_2T_2$.

A.2 Multistable Switch

Having constructed a bistable switch, we use induction to generalize our method to arbitrarily long sequences of target positions.

We assume we are already given a framework consisting of $(i-1)$ flippable triangles $\{\Delta S_1S_2T_1, \dots, \Delta S_{2(i-1)-1}S_{2(i-1)}T_{(i-1)}\}$, which realizes i target positions through the movement of its target end effector T_1 , i.e., $\rho_1(T_1) = \mathbf{t}_1, \dots, \rho_i(T_1) = \mathbf{t}_i$. Here, each realization ρ_i captures the consequences of flipping the first $(i-1)$ triangles across their respective fixed edges.

Now we aim to add another flippable triangle such that the framework is able to realize an additional target position \mathbf{t}_{i+1} by a flip of this triangle—that is, $\rho_{i+1}(T_1) = \mathbf{t}_{i+1}$. As in the previous constructions, we achieve i -th target position \mathbf{t}_i by flipping $(i-1)$ triangles, we denote the last flipped triangle as $\Delta S_{2(i-1)-1}S_{2(i-1)}T_i$ with vertex T_i at position \mathbf{t}_i . Then the construction of the new flippable triangle is simplified to the construction of a bistable switch. With the same construction steps, we can construct the i -th flippable triangle $\Delta S_{2i-1}S_{2i}T_i$. Figure 16 (b) shows an example of constructing the second bistable switch for a 3-state multistable switch.

In order to make the first $(i-1)$ triangles move alongside the new flippable triangle, as illustrated in Figure 16 (c), we connect the fixed edge $S_{2(i-1)-1}S_{2(i-1)}$ to one of the flippable edges $S_{2i}T_i$ in the last flippable triangle $\Delta S_{2i-1}S_{2i}T_i$ with three edges $S_{2(i-1)-1}S_{2i}, S_{2(i-1)}S_{2i}$ and T_iS_{2i} . In this way, the constructed switch structure is still a rigid graph, which ensures the topological rigidity of the entire multistable structure. Figure 16 (d), (e) shows the other two stable configurations of the 3-state multistable switch.

Finally we construct a physical compliant bar-joint mechanism based on the multistable switch graph by turning the flippable edges into compliant bars and keeping the other edges as rigid bars, please see Figure 16 (f). Note that in the set of connecting edges $\{S_{2(i-1)-1}S_{2i}, S_{2(i-1)}S_{2i}, T_iS_{2(i-1)}\}$, edge $S_{2(i-1)}S_{2i}$ and edge $T_iS_{2(i-1)}$ are omitted by fusing edges $S_{2(i-1)}S_{2(i-1)-1}$, $S_{2(i-1)-1}S_{2i}$ and $S_{2i}T_i$ at $S_{2(i-1)-1}$ and S_{2i} . This post-processing does not break the rigidity of the connection and reduces the total layers required for the switch structure.