DOI: 10.1111/cgf.13928
EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)

Volume 39 (2020), Number 2

Invertible Paradoxic Loop Structures for Transformable Design

Z. Li[†‡1], G. Nawratil[†2], F. Rist[3], M. Hensel[4]

[1]Johannes Kepler University Linz, Research Institute for Symbolic Computation, Austria
[2]TU Wien, Institute of Discrete Mathematics and Geometry & Center for Geometry and Computational Design, Austria
[3]TU Wien, Institute of Art and Design & Center for Geometry and Computational Design, Austria
[4]TU Wien, Institute of Architectural Sciences & Center for Geometry and Computational Design, Austria

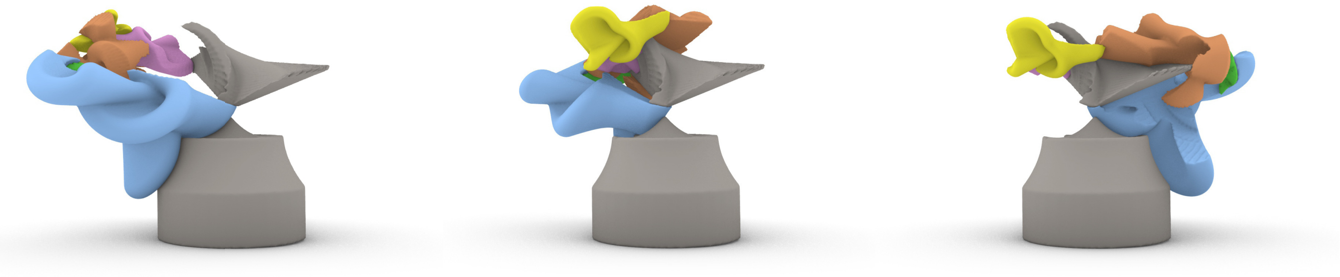


Figure 1: *Our interactive tool allows the user to design invertible paradoxic 6R loops, which are free of self-collisions.*

Abstract

We present an interactive tool compatible with existing software (Rhino/Grasshopper) to design ring structures with a paradoxic mobility, which are self-collision-free over the complete motion cycle. Our computational approach allows non-expert users to create these invertible paradoxic loops with six rotational joints by providing several interactions that facilitate design exploration. In a first step, a rational cubic motion is shaped either by means of a four pose interpolation procedure or a motion evolution algorithm. By using the representation of spatial displacements in terms of dual-quaternions, the associated motion polynomial of the resulting motion can be factored in several ways, each corresponding to a composition of three rotations. By combining two suitable factorizations, an arrangement of six rotary axes is achieved, which possesses a 1-parametric mobility. In the next step, these axes are connected by links in a way that the resulting linkage is collision-free over the complete motion cycle. Based on an algorithmic solution for this problem, collision-free design spaces of the individual links are generated in a post-processing step. The functionality of the developed design tool is demonstrated in the context of an architectural and artistic application studied in a master-level studio course. Two results of the performed design experiments were fabricated by the use of computer-controlled machines to achieve the necessary accuracy ensuring the mobility of the models.

CCS Concepts

• ***Theory of computation*** → *Computational geometry; Algorithmic mechanism design;* • ***Applied computing*** → *Computer-aided design;* • ***Computing methodologies*** → *Motion processing;*

1. Introduction

This research project was inspired by the *invertible cube* (Fig. 2) of Paul Schatz (1898–1979) [Sch13], who was an artist, inventor, and technician trying to bridge the gap between art and natural sciences. This so-called *Schatz linkage* consists of six parts of a cube, which are connected by rotary joints (R-joints) to form a closed chain. Its rhythmic and aesthetic mobility was the basis upon which Schatz built his later work, leading also to practical applications such as the *oloid*-agitator or the mixing device *Turbula*.

From the kinematic point of view, the *Schatz linkage* is a closed serial 6R loop. In general, such a ring structure possesses a finite number of poses (at most 16). Only if the linkage geometry

[†] Joint first author
[‡] Corresponding author

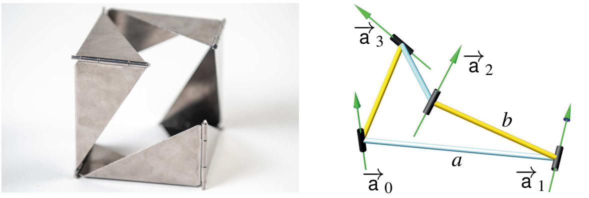


Figure 2: *(left) Schatz linkage (right) Bennett mechanism: The common normals of adjacent (oriented) rotation axes $\vec{a}_i$ and $\vec{a}_{i+1}$ (for $i = 0,\dots,3 \bmod 4$) form a closed polyline, where opposite sides have equal lengths a and b. Moreover for $\theta_i := \angle(\vec{a}_i, \vec{a}_{i+1})$, the following three conditions have to hold; (1) $\theta_0 = \theta_2$, (2) $\theta_1 = \theta_3$ and (3) $a \sin\theta_1 = b \sin\theta_0$.*

meets certain constraints does the structure become mobile, and the resulting mechanism as well as its associated motion is called *paradoxic*. The first 6R loop with a paradoxic mobility dates back to Sarrus [Sar53], which has many applications (e.g. retractable mechanism of awnings). Up to now, numerous examples have been found [Li15,CY12,Die95] but the classification of these 6R linkages is still open.

Transformable Design While the geometry of mechanisms has so far mainly been used for ensuring the desired functionality of that mechanism (e.g. achieving special paths for certain points on the end-effector, see [MS11]), the appearance of that mechanism as it changes over time has not been a topic of interest, apart from a few exceptions in the context of kinetic art, e.g. the well-known expanding structures of Chuck Hoberman [HAI90]. Since his pioneering work on transformable design, the interest and developments in this topic have increased in recent decades due to new applications in engineering (e.g. tensegrity bridges [PTV*17]), robotics (e.g. deployable mechanisms [ZSC16]), material sciences (e.g. reconfigurable metamaterials [OWHB17]), medicine (e.g. auxetic stents [KLPCP18]), etc. A further creative discipline – beside kinetic art – taking the aesthetic transformation of structures into account, is architecture; especially of interest are adaptive facade applications mostly under the functional aspects of shading/lighting [PEVW15,Mar16,BLZM16]. It is beneficial to base those shading systems on mechanisms with a 1-parametric mobility (shading in dependence on the time t of day), as their control requires only one active joint. An additional nice feature of the 6R loops treated in the paper, beside the fact that R-joints are cheap to produce and easy to maintain, is that the rotation angle of each R-joint is strictly increasing during the motion, thus the single motor can be located in any of the six axes (preferably in one of the two axes belonging to the resting link). This has a positive impact on the cost, weight, and design of the shading elements.

Goal & Overview The goal of this paper is to open a new design space for paradoxic closed 6R chains from the perspective of transformable design, focusing on architectural and artistic applications. We present an interactive design tool compatible with existing software that allows non-expert users to create these invertible loops, which are self-collision-free over the complete motion cycle. It should be pointed out that our plug-in does not generate these transformable structures fully automatically, but it provides several user interactions that facilitate design exploration. This allows users freedom to tap into their creative potential to make decisions that are hard to formalize and quantify.

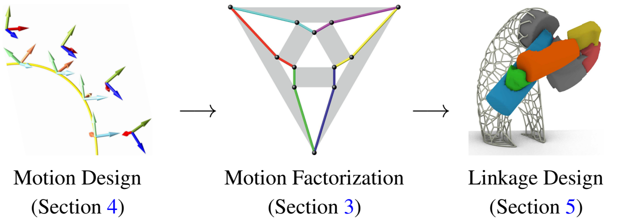


Figure 3: *The design tool's workflow can roughly be divided into three steps. A more detailed flowchart is given later on in Fig. 4.*

Note that our study goes along with the recently increasing interest in the computational design of physical mechanical models indicated by the publications [ZSC16,PTV*17,CTN*13,TCG*14, MZB*17,YCC17,YZC18,ZAC*17,NBA19,GJG16].

The paper is structured as follows: After a review of related work in Sec. 2, we explain the kinematic concepts the paper is based on in Sec. 3. The aspects of motion and linkage design are studied in Sec. 4 and Sec. 5, respectively. The article is closed by results and their discussion in Sec. 6. Note that the structure of the paper does not follow the rough workflow of the presented tool given in Fig. 3 for reasons of readability and clarity. A first insight into the complete design tool workflow can be gained by the detailed flowchart given in Fig. 4, whose individual steps will be explained in the subsequent sections.

2. Related Work and Contributions

Transformable Structures in Architecture/Design/Art According to Tachi [Tac10], transformable freeform surfaces can be approached by foldable rigid origami, which goes beyond regular folding patterns (such as Miura-ori used for e.g. solar panels). Moreover, the folding motion and its actuation is also part of the design problem [TH17]. All these tasks are much more challenging for curved origami [RHSH18, KMM17], possessing great design potential, as well as Kirchhoff-Plateau surfaces, which are planar rod networks embedded in pre-stretched fabric that deploy into complex 3-dimensional shapes [POT17]. Another interesting approach towards the topic of transformable surfaces is programmable auxetics [KLPCP18], which can snap between multiple stable configurations [SLRP18]. In this context also HYBRIDa's Hypermembrane DHUB [FT13] should be noted, which is a shape-adaptable self-supporting structure able to stand in different equilibrium positions, where a bendable grid structure covered by elastic materials is flexed by a few prismatic actuators. Recently Panetta et al. [PKLI*19] studied similar structures having the property that the underlying grid possesses a flat configuration in the relaxed state, which can be assembled from linear beam elements coupled through rotational joints. Barton et al. [BSK*13] dealt with transformable structures composed of a series of flexible 6-snakes. Each snake of smoothly joined circular arcs can be seen as a 7R closed linkage with coplanar adjacent neighboring R-joints (with the exception of the two

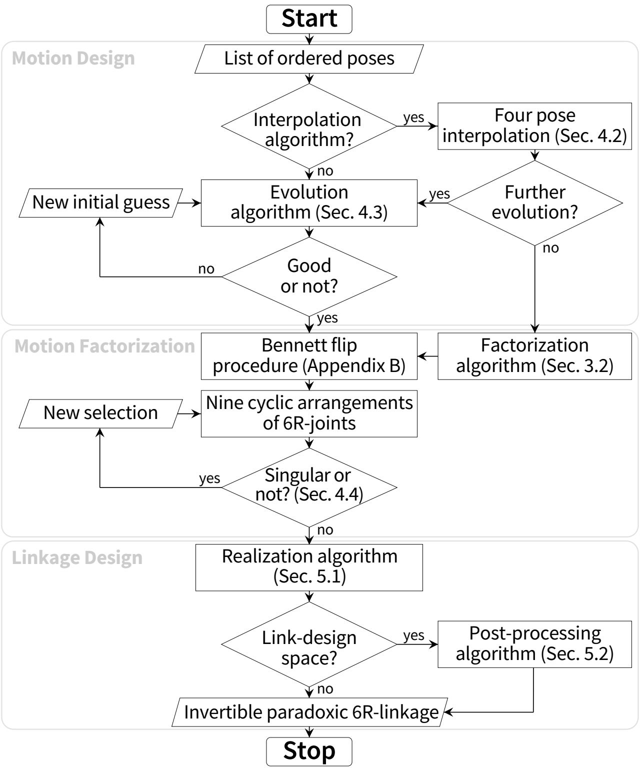

Start
Motion Design
List of ordered poses
Interpolation algorithm?
yes
Four pose interpolation (Sec. 4.2)
no
New initial guess
Evolution algorithm (Sec. 4.3)
yes
Further evolution?
no
no
Good or not?
yes
Motion Factorization
Bennett flip procedure (Appendix B)
Factorization algorithm (Sec. 3.2)
New selection
Nine cyclic arrangements of 6R-joints
yes
Singular or not? (Sec. 4.4)
no
Linkage Design
Realization algorithm (Sec. 5.1)
Link-design space?
yes
Post-processing algorithm (Sec. 5.2)
no
Invertible paradoxic 6R-linkage
Stop

Figure 4: *A detailed flowchart of the design tool's workflow.*

R-joints fixed to the ground). Note that corresponding circular arcs of neighboring snakes cannot be connected with rigid surface elements (but e.g. with elastic membranes), as their relative position changes over the time of transformation. Textile/fabric foldings were already used by the architect Perez-Pinero for covering the roof of his famous movable theater from 1961 [Esc13], which was the starting point for the development of deployable bar-structures based on scissor-like elements furthered by e.g. Calatrava [Cal81] and Hoberman [HAI90]. The cutting edge in this field is represented in [ZSC16]. Two special types of deployable ring structures (*regular-polygonal* and *doubly symmetric* rings) with more than six R-joints were studied by Viquerat and Guest [VG13]. The collision-free unfolding of planar closed linkages can be realized by an energy-driven approach [CDIO04], which can also be used for a chain-based physical transformation between 3D models [YYL*19].

Further examples of transformable loop structures are as follows: *Schatz linkages* were proposed for the design of pop-up stores [Gai15] as well as facade applications [sch04,EWMH14]. The *invertible cube* can also be seen as a special dimensioned *kaleidocycle*, which are closed chains of hinged tetrahedra [SW77]. These invertible rings trace back to the graphic designer Walker [Wal67], and they are still a topic of recent research [KKP19]. The mathematical loop with the most impact in the art/design community is the *Möbius strip* [Pic06], whose kaleidocyclic realization was presented recently [SF19]. Furthermore, we want to point out Len Lye's kinetic work of art that incorporates winding loops of steel generating organic shapes [LLF80]. Clearly our project is also related to the field of linkage based kinetic sculptures containing closed kinematic loops. In this context, we only want to name Theo Jansen [TJS90], who is probably the best known contemporary representative due to his "Strandbeest".

Paradoxic Loops A closed nR chain consists of n rigid bodies (links) as well as R-joints that connect the links cyclically together. According to the formula of Chebychev-Grübler-Kutzbach [MS11] such a mechanical loop has mobility $n-6$. Therefore mobile nR loops with $n<7$ are called paradoxic. Mobile 3R loops do not exist, and the synthesis classification of paradoxic 4R loops has only one spatial linkage, which is the so-called Bennett mechanism [Ben03] explained in (Fig. 2). The Goldberg linkage [Kar98] is the only paradoxic spatial 5R linkage, which can be constructed by merging two Bennett mechanisms [Gol43].

Since the previously mentioned linkage of Sarrus [Sar53], numerous paradoxic 6R loops were discovered [CY12,Die95,Li15] but a full listing is still to be found. Among all known mobile 6R linkages, a large number results from combining Bennett or Goldberg linkages. In contrast to this traditional way of synthesis [CY12,Die95], we use the completely different method of *motion factorization*. Roughly speaking this procedure introduced by Hegedüs et al. [HSS13a] consists of three steps; (i) starting with a rational motion, (ii) factoring it into two different ways, which correspond to two open serial chains, and (iii) combining them to a closed linkage. A lot of already known paradoxic loops can be constructed in this way [CY12,Die95], like the Bennett mechanism (rational quadratic motion [BSH05]) or the type III Bricard linkage [HSS13a] (rational cubic motion).

Note that a variation of this method can be used to synthesize a mechanism which can follow any given rational curve in 3-space [LSS18], where those with a straight line trajectory have been studied in more detail [LSS16b]. Moreover, Rad and Schröcker [RS18] combined the motion factorization approach with curve evolution methods to generate paradoxic 6R loops that approximate a set of given target poses.

Contributions As this paper presents for the first time the method of motion factorization to the computer graphics community, great effort is put in the preparation of the underlying and related kinematic and mathematical concepts. Besides the development of the interactive tool compatible with Rhino/Grasshopper to design invertible paradoxic 6R loops in the context of architectural and artistic applications, the paper also contains the following scientific contributions:

- Until now there are no results concerning the existence of spatial paradoxic loops, which are collision-free over the complete motion circle. Our obtained statistical results (Sec. 6.3) imply the conjecture that a general rational cubic motion can always be realized by a collision-free 6R loop.
- We presented a novel method for generating link-design spaces (Sec. 5.2) which can also be used for solving the loop grounding problem.
- The several improvements/modifications of the *evolution algorithm* of Rad and Schröcker [RS18] (Sec. 4.3) were driven by our interest in shaping the complete motion and not only a part of it. Therefore the evolution process is split up into three stages

which differ in the selection strategy of the guiding poses, the parametrization of the linear motion polynomials and the weighting of the orientation and translation component of the used object dependent metric.

- Further minor scientific contributions are the (a) Bennett flip procedure (Appendix B) to obain another factorization instead of rerunning the full factorization algorithm of [HSS13a], and numerical implementation of the algorithm of [HSS13a] in *C#*. (b) motion selection criteria of the four pose interpolation as well as its interactive (space mouse) implementation (Sec. 4.2) and (c) indication of proximity to singular rational cubic motions (Sec. 4.4).

3. Kinematic Concepts

Dual quaternions introduced by Clifford [Cli71] are well-known to the computer graphic community in the context of skinning (e.g. [KCvO08, LH16]). A further application mentioned by Hanson [Han12] is the motion interpolation problem arising in computer animation (either for the movement of objects or the motion of the camera) for which sophisticated algorithms [GR94, PG05, Naw19] were developed by kinematicians in the context of mechanism science. In the latter field, dual quaternions are well established due to the fundamental works of Study [Stu91] and Kotelnikov [Kot95] and the persistent reanimation of this powerful tool by Dimentberg [Dim48], Blaschke [Bla60] and Yang [Yan63].

We proceed with a brief introduction into the dual-quaternion representation of spatial displacements (Sec. 3.1), which is the basis of the *motion factorization* method (Sec. 3.2) used for the synthesis of transformable 6R loops.

3.1. Dual-Quaternion Representation

Notation Convention We denote the skew field of quaternions by $\mathbb{H}$ and its elements by

$$\mathfrak{Q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad \text{with} \quad q_0, \ldots, q_3 \in \mathbb{R}, \tag{1}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the usual quaternion units. The quaternion $\mathfrak{Q}$ can be split into a scalar part q_0 and a vector part $\mathfrak{q} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, thus we can write $\mathfrak{Q} = q_0 + \mathfrak{q}$. The quaternion conjugation is denoted by $\widetilde{\mathfrak{Q}} = q_0 - \mathfrak{q}$. If the quaternion product $\mathfrak{Q}\widetilde{\mathfrak{Q}} = 1$ holds, $\mathfrak{Q}$ is called a unit-quaternion.

Moreover we need the ring $\mathbb{D}$ of dual numbers $\underline{q} = p + \varepsilon d$ where the primal part p as well as the dual part d are reals. The so-called dual-unit $\varepsilon \neq 0$ has the property $\varepsilon^2 = 0$. Every analytic function f can be extended to dual arguments as follows:

$$f(\underline{q}) := f(p) + \varepsilon d f'(p). \tag{2}$$

The ring of dual quaternions is denoted by $\mathbb{DH}$ and its elements by $\underline{\mathfrak{Q}} = \mathfrak{P} + \varepsilon\mathfrak{D}$ with $\mathfrak{P}, \mathfrak{D} \in \mathbb{H}$. Its conjugation is given by $\widetilde{\underline{\mathfrak{Q}}} = \widetilde{\mathfrak{P}} + \varepsilon\widetilde{\mathfrak{D}}$.

Spatial Displacement $\underline{\mathfrak{Q}} = \mathfrak{P} + \varepsilon\mathfrak{D}$ is a so-called unit dual quaternion if $\underline{\mathfrak{Q}}\widetilde{\underline{\mathfrak{Q}}} = 1$, which is the case iff $\mathfrak{P}$ is a unit-quaternion and the so-called Study condition

$$\mathfrak{P}\widetilde{\mathfrak{D}} + \mathfrak{D}\widetilde{\mathfrak{P}} = 0 \tag{3}$$

holds. It is well known (e.g. [BR79]) that every unit dual quaternion represents a spatial displacement (cf. Appendix A) and that it can be rewritten in the form:

$$\cos\tfrac{\underline{\alpha}}{2} + \sin\tfrac{\underline{\alpha}}{2}\,\underline{\mathfrak{a}} \quad \text{with} \quad \underline{\mathfrak{a}}\widetilde{\underline{\mathfrak{a}}} = 1, \tag{4}$$

where the trigonometric functions of the so-called dual angle $\underline{\alpha}$ are computed according to Eq. (2). This is a very nice representation as it provides direct access to the geometry of the displacement in the sense of Chasles' theorem stating that any two poses of a moving system in 3-space can be transformed into each other by a screw displacement: The primal part of $\underline{\mathfrak{a}}$ is a unit-vector indicating the direction of the screw axis $\vec{a}$ (an oriented line), and the dual part its moment vector. The angle of rotation about $\vec{a}$ is given by the primal part of $\underline{\alpha}$, and the translation distance along $\vec{a}$ by the dual part of $\underline{\alpha}$. The latter vanishes for the case of a pure rotation where $\underline{\alpha}$ can be replaced by α in Eq. (4).

Kinematic Mapping The set of unit dual quaternions $\underline{\mathfrak{Q}}$ constituting the dual unit-sphere in $\mathbb{D}^4$, yields a double cover of the Euclidean motion group SE(3), which can be avoided by identifying their antipodal points $\pm\underline{\mathfrak{Q}}$ by means of homogeneous coordinates $(p_0 : p_1 : p_2 : p_3 : d_0 : d_1 : d_2 : d_3)$, also known as Study parameters. Then there is a bijection ψ between SE(3) and all points of $\mathbb{RP}^7$ located on the Study quadric Ψ given by Eq. (3) without including the 3-dimensional generator space G: $p_0 = p_1 = p_2 = p_3 = 0$.

Points of the ambient space can be projected onto the Study quadric with a mapping studied in [PSH18, SWC18, Naw19]. This projection φ: $\mathbb{RP}^7 \setminus G \to \Psi \setminus G$ maps a dual quaternion $\mathfrak{P} + \varepsilon\mathfrak{D}$ with $\mathfrak{P}\widetilde{\mathfrak{P}} = 1$ to the following unit dual quaternion:

$$\mathfrak{P} + \varepsilon\left[\mathfrak{D} - \tfrac{1}{2}\left(\mathfrak{D}\widetilde{\mathfrak{P}} + \mathfrak{P}\widetilde{\mathfrak{D}}\right)\mathfrak{P}\right]. \tag{5}$$

3.2. Factorization of Rational Motions

In the following, we study rational motions of the end-effector and their different factorizations, where each linear term of a factorization corresponds to a transformation applied by one link of the associated serial chain. Finally, we discuss for the cubic case the possible combinations of the resulting open 3R chains for generating a paradoxic 6R loop.

Rational Motions Rational motions [Rös98, JW02] are defined by the property that the trajectory of every point of the moving space is a rational curve.

According to Jüttler [Jüt93] every rational motion can be represented by a rational curve on the Study quadric Ψ. As a consequence (Sec. 3.1) every rational motion corresponds to a polynomial dual quaternion $\underline{\mathfrak{M}}(t) \in \mathbb{DH}(t)$ with a (nonzero) norm polynomial $\underline{\mathfrak{M}}(t)\widetilde{\underline{\mathfrak{M}}}(t) \in \mathbb{R}[t]$, where the leading coefficient is invertible (otherwise, one needs a reparametrization) and it is on the Study quadric Ψ by [LSS16a]. Such a dual quaternion polynomial $\underline{\mathfrak{M}}(t)$ is called a *motion polynomial*, where the variable t can be interpreted as time. Dividing the motion polynomial by the leading coefficient yields a monic motion polynomial whose motion differs from the initial motion only by a fixed transformation (the leading coefficient). For a monic motion polynomial, the leading coefficient corresponds to the identity transformation where $t = \infty$. Therefore, it is enough to consider the monic case for simplicity.

Motion Factorization According to Hegedüs et al. [HSS13a] a generic monic motion polynomial $\underline{\mathfrak{M}}(t)$ of degree n admits at most $n!$ factorizations of the shape

$$(t-\underline{\mathfrak{A}}_0)\dots(t-\underline{\mathfrak{A}}_{n-1}) \quad \text{with} \quad \underline{\mathfrak{A}}_i = a_i + \underline{\mathfrak{a}}_i \quad \text{and} \quad a_i \in \mathbb{R} \qquad (6)$$

for $i = 0,\dots,n-1$, where the term "generic" means that the primal part of $\underline{\mathfrak{M}}(t)$ has no real polynomial factors. As the ith factor gives a rotation about the axis $\overrightarrow{\underline{\mathfrak{a}}}_i$, where a_i corresponds to an angular shift (Eq. (4)), each factorization describes a decomposition into serial rotations, which can be realized by an open serial nR-chain. Due to Eq. (6) and the parameter $t \in (-\infty,\infty)$, the relative motion of two links can reach a full turn. Moreover the angle of this relative rotation as a function of the time t is strictly increasing, which means that a link cannot stop or change its rotational direction.

The factorization (6) can be computed by an extended version of the Euclidean algorithm [HSS13a]. The flow of this recursive algorithm is as follows: a real quadratic factor of the norm polynomial is taken to divide the motion polynomial, the remainder gives a linear dual quaternion polynomial whose right root implies a right root of the motion polynomial regarding the chosen real quadratic factor. The details of this algorithm are explained in Hegedüs et al. [HSS13a] but a few comments regarding our *C#* implementation for Rhino/Grasshopper are listed:

- The *MathNet* library is used for the numeric univariate polynomial remainder and division calculation.
- Real quadratic factors of the monic norm polynomial are computed indirectly; all complex solutions are calculated numerically and then a real quadratic factor can be obtained from two very close complex solutions.
- The implementation of the dual quaternion computation is based on the quaternion computation in Rhino/Grasshopper.

Quadratic Motion and the Bennett Mechanism It is well-known [Ham11] that the motion of the Bennett mechanism (Fig. 2) corresponds to a conic section of the Study quadric Ψ; thus it can be synthesized by three prescribed poses [BSH05]. As a consequence, this quadratic motion has two different factorizations which give us the two 2R-chains yielding the Bennett mechanism [HSS13a]. For instance, the product $(t-\mathbf{i})(t-2\mathbf{j}-\varepsilon\mathbf{i})$ can also be factorized as

$$(t-\tfrac{8}{5}\mathbf{i}-\tfrac{9}{25}\varepsilon\mathbf{i}-\tfrac{6}{5}\mathbf{j}+\tfrac{12}{25}\varepsilon\mathbf{j})(t+\tfrac{3}{5}\mathbf{i}-\tfrac{16}{25}\varepsilon\mathbf{i}-\tfrac{4}{5}\mathbf{j}-\tfrac{12}{25}\varepsilon\mathbf{j}).$$

Cubic Motion and 6R Loops In the remainder of the paper, we focus on rational cubic motions as they imply paradoxic 6R loops. As there exist 3! factorizations, the 6R loop is not uniquely determined. Combining two factorizations produces a 6R loop if the first and last factors are not the same; i.e.

$$(t-\underline{\mathfrak{A}}_0)(t-\underline{\mathfrak{A}}_1)(t-\underline{\mathfrak{A}}_2) = (t-\underline{\mathfrak{A}}'_0)(t-\underline{\mathfrak{A}}'_1)(t-\underline{\mathfrak{A}}'_2), \qquad (7)$$

with $\underline{\mathfrak{A}}_0 \neq \underline{\mathfrak{A}}'_0$ and $\underline{\mathfrak{A}}_2 \neq \underline{\mathfrak{A}}'_2$. Therefore each factorization can only be combined with three (out of five) factorizations to form a 6R loop. In total, there are 9 different 6R loops (Fig. 5) belonging to two different types: *angle-symmetric 6R loops* (third type in Li et al. [LS13]) and *double Bennett 6R loops* [Die95].

Note that we only need to calculate one factorization of a cubic motion, as the (at most) other five factorizations can be obtained by a Bennett-flip procedure [LSS18] due to Theorem 1 of Appendix B. For a cubic motion polynomial, all possible Bennett flips are revealed in Fig. 5. For instance, after a Bennett flip, the red chain becomes the green chain. A further flip changes the green chain to the blue one which forms a *double Bennett 6R linkage* with the red chain. One more flip and the blue chain becomes the yellow one that forms an *angle-symmetric 6R linkage* with the red chain. In this way, one can reduce computational costs as only 4 (*double Bennett 6R loop*) or 6 (*angle-symmetric 6R loop*) quaternion multiplications have to be performed instead of 7 within the factorization algorithm.

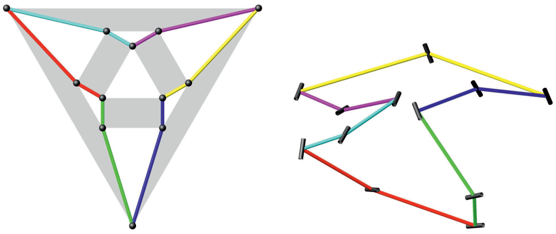


Figure 5: *(left) Schematic arrangement of the six serial chains (colored in red, green, blue, yellow, purple, cyan): The vertices of the graph correspond to rotation axes and the edges to links. The gray quads indicate that the involved four axes generate a Bennett mechanism. Nine possible 6R loops can be generated by combining two chains, which are not allowed to share a common rotation axis. (right) Illustration of all the six serial chains for a concrete example.*

4. Motion Design

In this section, we discuss two design possibilities of a rational cubic motion, which is the input (Fig. 3) for the motion factorization (Sec. 3.2) and in series for the linkage design (Sec. 5). For four given poses, a cubic motion interpolation (Sec. 4.2) can be performed. If more than four poses are given, we have to find a cubic motion by means of curve evolution (Sec. 4.1) approximating the given data best in the sense of a metric discussed next.

4.1. Metric

It is well known [MSZ94] that there does not exist a (positive-definite) metric on SE(3) that is invariant with respect to changes of the fixed frame and the moving frame, respectively. Due to Park [Par95], there is an approach to come up with a geometrically meaningful distance function by considering the distance between two poses σ_1 and σ_2 of the same rigid body (e.g. a shading element), which yields a so-called *object dependent metric* firstly studied by Kazerounian and Rastegar [KR92]. Assuming uniform mass distribution, their metric can be simplified to [PHR04]:

$$\mathrm{dist}^2 := \frac{1}{6}\sum_{i=1}^{6} \|\sigma_1(\mathbf{v}_i)-\sigma_2(\mathbf{v}_i)\|^2, \qquad (8)$$

where $\mathbf{v}_i$ $(i=1,\dots,6)$ are the six vertices of the object's inertia ellipsoid (centered on the barycenter). In some situations, we scale this ellipsoid with a scaling factor in the interval $]1;\infty[$ or $]0;1[$ in order to give more weight on the orientation or translation component, respectively. The inertia ellipsoid is illustrated in the plug-in to enhance the user's perception of the distance function.

4.2. Four Pose Interpolation

In [HSS15], a framework of synthesizing cubic motions which can visit four given poses was presented. Based on the kinematic mapping ψ at the end of Sec. 3.1, its construction can easily be explained geometrically. In the general case, the four points on the Study quadric Ψ, which correspond to the given poses, span a projective 3-space (otherwise no interpolating cubic motion exists) intersecting Ψ along a ruled quadric Φ. For construction of the cubic, we are aiming a second quadric Λ in this 3-space which has a ruling with Φ in common as well as the four given poses. For each ruling, there exists a bundle of such quadrics Λ, which intersect Φ along the same cubic, thus there is a bijection between rulings and cubics. For details of the parametrization of the resulting two families of cubics (which is based on a standard interpolation method), we refer to [HSS15].

Motion Selection In most applications, the visit order of the four given poses is crucial. In [HSS15], it is also mentioned that within each family, the visit order is the same. If this order is not as expected, one has to change the family or in the worst case the four given poses. If the order is correct, the user can pick out a suitable cubic according to the following exemplary criteria:

- Distance of the rotation axes associated with the end-link (axes $\vec{a}_2$ and $\vec{a}_3$) and/or the base-link (axes $\vec{a}_0$ and $\vec{a}_5$) from some user-defined locations.
- Length of the motion according to the above metric (alternatively, one can also evaluate the length of an end-effector trajectory).
- Shape of the motion by adding further poses and picking the cubic motion which is closest to these poses in the sense of the metric (8). This criterion is of interest for generating an initial guess for the motion evolution discussed in Sec. 4.3.

Clearly the user can also consider other criteria fitting better for their task at hand, which can be formulated mathematically in terms of a cost function, and optimize them by using the Rhino/Grasshopper tool *Galapagos*.

Our implementation also allows fixing only three input poses of the interpolation and controlling the fourth pose interactively by means of a space mouse. This approach can be very well integrated into the designer/architect's workflow.

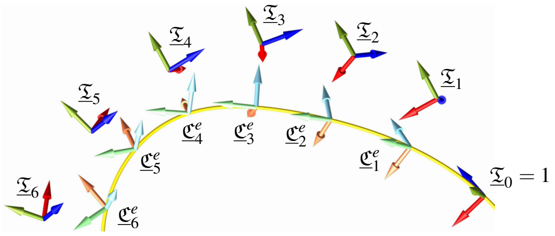
4.3. Motion Evolution

Our goal is to find a rational rigid body motion (a rational curve on the Study quadric Ψ) via a curve evolution procedure without taking into account dynamical constraints [CAL17]. Curve evolution for fitting a parametrized curve to a given set of points is an important tool in geometric modeling and computer vision. A hybrid curve fitting algorithm was developed by Aigner and Jüttler [AJ07] to find an interpolation among a chosen family of planar curves for approximating a given unorganized (without order) point cloud. This so-called curve evolution procedure works as follows: Normal velocities of the closest points to the given ones are used to guide an iteration procedure for the parameters of the curve family. The iteration direction of the parameters is obtained by a least-squares solution and the associated step size is user-defined.

In [RS18, Rad18] this curve evolution procedure was adopted for motion approximation of an ordered set of target poses. They considered the family of cubic rational curves on the Study quadric $\Psi \in \mathbb{RP}^7$ parametrized as a product of three linear motion polynomials, where each is of the form:

$$t - x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + \\ \varepsilon((x_2x_7 - x_3x_6)\mathbf{i} + (x_3x_5 - x_1x_7)\mathbf{j} + (x_1x_6 - x_2x_5)\mathbf{k}),$$

which is redundant as the set of pure rotations is a 5-dimensional variety in $\mathbb{RP}^7$. Driven by our interest in shaping the complete motion and not only a part of it as done in [RS18, Rad18], we modified Rad and Schröcker's algorithm by splitting up the evolution process into three stages which differ in the selection strategy of the guiding poses, the parametrization of the linear motion polynomials and the weighting of the orientation and translation component of the metric. In this context, the loop phenomenon may arise (Fig. 17), which can also be handled by the given algorithm.



$\underline{\mathfrak{T}}_3$
$\underline{\mathfrak{T}}_4$
$\underline{\mathfrak{T}}_2$
$\underline{\mathfrak{T}}_1$
$\underline{\mathfrak{T}}_5$
$\underline{\mathfrak{C}}_4^e$
$\underline{\mathfrak{C}}_3^e$
$\underline{\mathfrak{C}}_2^e$
$\underline{\mathfrak{C}}_5^e$
$\underline{\mathfrak{C}}_1^e$
$\underline{\mathfrak{T}}_0 = 1$
$\underline{\mathfrak{T}}_6$
$\underline{\mathfrak{C}}_6^e$

Figure 6: *The target poses $\underline{\mathfrak{T}}_0, \ldots, \underline{\mathfrak{T}}_6$ and the guiding poses $\underline{\mathfrak{C}}_0^e, \ldots, \underline{\mathfrak{C}}_6^e$ are illustrated via orthogonal frames. In addition, the origin's trajectory under the rational cubic motion $\underline{\mathfrak{C}}^e(t)$ is drawn.*

Evolution Algorithm In the first step, the user has to select one target pose that is most important to be hit exactly. By a change of the reference frame, we can assume without loss of generality that this selected target pose equals the identity transform. Moreover, we assume that our task necessitates an ordering of the m target poses; which are denoted by $\underline{\mathfrak{T}}_0 = 1, \underline{\mathfrak{T}}_1, \ldots, \underline{\mathfrak{T}}_{m-1}$ (Fig. 6). If the visit order is unspecified, the problem simplifies, and the algorithm given next can easily be cut down by the reader/user.

(1) Initial guess: If no good initial guess is known, we start with a spherical motion parametrized by the cubic motion polynomial

$$\underline{\mathfrak{C}}^0(t) = (t - r_0 + \cdots + r_3\mathbf{k})(t - r_4 + \cdots + r_7\mathbf{k})(t - r_8 + \cdots + r_{11}\mathbf{k}), \quad (9)$$

where $r_0, \ldots, r_{11}$ are random real numbers. Further cubic motions obtained during the evolution are denoted by $\underline{\mathfrak{C}}^e(t)$ with $e \in \mathbb{N}$.

(2) Guiding poses: An important ingredient is the selection of the so-called guiding poses $\underline{\mathfrak{C}}_i^e := \underline{\mathfrak{C}}^e(t_i)$ for $i = 1, \ldots, m-1$ (Fig. 6), which are used to evolve the motion iteratively to the corresponding target poses. We apply one of the following two methods:

(a) *Closest pose projection:* We compute the guiding pose $\underline{\mathfrak{C}}_i^e$ as the one closest to the target pose $\underline{\mathfrak{T}}_i$ in sense of the metric (8). This corresponds to the initial approach of [AJ07], which was also used in [RS18, Rad18]. It turns out that the local extrema of the distance function can be obtained as the zeros of a univariate polynomial in t of degree at most 10, which was also mentioned in [RS18, Rad18].

(b) *Proportionally spaced:* We compute the guiding points $\underline{\mathfrak{C}}^e(t_i)$ in a way that the following relation is approximately fulfilled:

$$\overline{\underline{\mathfrak{T}}_{i-1}\underline{\mathfrak{T}}_i} : \widehat{\underline{\mathfrak{C}}^e_{i-1}\underline{\mathfrak{C}}^e_i} = \overline{\underline{\mathfrak{T}}_i\underline{\mathfrak{T}}_{i+1}} : \widehat{\underline{\mathfrak{C}}^e_i\underline{\mathfrak{C}}^e_{i+1}} \qquad (10)$$

for $i = 0,\dots,m-1$ (i mod m), where the overline-bar denotes the distance (8) and the hat-sign indicates the approximate lengths along the motion based on an approximate discretized arc-length segmentation of $\underline{\mathfrak{C}}^e(t)$, i.e., the time interval is discretized into a user-defined number of time instances in a way that the distance (8) between two adjacent time instances is constant. The starting point for the layout of the guiding poses is $\underline{\mathfrak{T}}_0 = 1$ and the sampling direction is determined by placing $\underline{\mathfrak{C}}^e_1$ on the same side of the motion as its closest pose to $\underline{\mathfrak{T}}_1$.

(3) Curve Evolution: The strategy for the evolution process splits up into the following three stages:

- *Initial stage:* We put a high weight (user-defined value > 1) on the orientation part of the metric (Sec. 4.1) in this stage, due to the observation of Rad and Schröcker [RS18] that their results fit much better in position than orientation. With respect to this weighted metric, we compute the guiding poses according to approach (2b). Moreover, we parametrize each of the three linear factors in the following form:

  $$t - x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + \varepsilon(x_5\mathbf{i} + x_6\mathbf{j} + x_7\mathbf{k})$$

  The iteration of the parameters is given by $x_i \mapsto x_i^* := x_i + s\Delta x_i$ where $s \in]0,1]$ denotes the step size and the Δx_i is obtained by applying the Gauss-Newton method to minimize the nonlinear least squares objective of fitting the target poses. The only difference to [RS18, Rad18] is the parametrization. In general, the updated 8-tuple $(t - x_0^* : x_1^* : x_2^* : x_3^* : 0 : x_5^* : x_6^* : x_7^*)$ does not fulfill the Study condition (3). Therefore we project it back onto the Study quadric Ψ by the map φ (5). This projection fits very well with our strategy of favoring the orientation part, as it only affects the parameters x_5^*, x_6^*, x_7^*, which control only the translation part of the spatial displacement. We proceed to the next stage as soon as the t-values of the closest poses (2a) follow the ordering of the target poses.
- *Middle stage:* Now the guiding poses are selected by approach (2a). The rest of the iteration remains the same as in the initial stage with the sole difference that we reduce the weight on the orientation successively by a fixed percentage (user-defined value) until we end up with the original metric. Once this point is reached and the improvement by the evolution steps is insignificant, we go on to the final stage.
- *Final stage:* In order to switch to a parametrization that automatically satisfies the Study condition (3) so that an optimality-sacrificing projection φ is not needed after each step we use one of the following three parametrizations

  $$t - x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + \varepsilon(x_5(x_2\mathbf{i} - x_1\mathbf{j}) + x_6(x_3\mathbf{i} - x_1\mathbf{k}))$$

  $$t - x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + \varepsilon(x_5(x_2\mathbf{i} - x_1\mathbf{j}) + x_6(x_3\mathbf{j} - x_2\mathbf{k}))$$

  $$t - x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + \varepsilon(x_5(x_3\mathbf{i} - x_1\mathbf{k}) + x_6(x_3\mathbf{j} - x_2\mathbf{k}))$$

  based on whether x_1, x_2, or x_3 has the largest magnitude, respectively. These parametrizations, which are not used in the earlier stages because this would make the minimization problem too nonlinear to solve efficiently, degenerate when the corresponding x_1, x_2, or x_3 value vanishes. In this stage, the guiding poses are selected by approach (2a) with respect to the original metric.

Comments on the Evolution Algorithm We experienced that our algorithm might need several attempts of random initial guesses (9) to ensure good convergence, which can be accomplished by employing *Galapagos* in Rhino/Grasshopper. This goes along with the observations of Rad and Schröcker [RS18]. Moreover, we note that the described algorithm cannot be performed in real-time, as the approximate discretized arc-length segmentation used for the guiding pose selection (2a) is time consuming. If a good approximating motion is already known, then our algorithm does not flow through this bottleneck; thus the motion can be modified interactively by applying small changes to the given target poses with a space mouse. Further comments are listed:

- *Initial guess:* As already mentioned in Sec. 4.2, the four pose interpolation can be used for generating a good initial guess.
- *Step size:* We use the Rhino/Grasshopper tool *Galapagos* for finding a good step size as it turned out that this approach is superior to the step size selection proposed in [RS18].
- *Order issue*: If the ordering of the closest poses is damaged during the middle or final stage we go back to the initial one. We set a maximal number of iterations to avoid infinite loops and proceed with a new random initial guess (9).
- *Reference frame:* In the final stage, one can also decouple the reference frame from the first target pose by left multiplying the three linear factors by a dual quaternion, whose entries are included in the iteration procedure. In this way, a fairer (i.e. without preferring any target pose) rational cubic approximation of the target poses can be found.

4.4. Singularities

As the studied 6R loops are of mobility one, their spatial shape can be transformed by actuating one R-joint only. But difficulties can arise in the motion transmission through the kinematic chain if so-called singular configurations are encountered. For the considered structures, these singularities are characterized by the line-geometric property that the six rotational axes belong to a linear congruence of lines [PW11]. As a singularity only depends on the rotation axes, its avoidance is a matter of the motion design instead of the linkage design (Sec. 5). The existence of these singularities is demonstrated by type III Bricard linkages (e.g. [Bak09]), which have two configurations of coplanar rotation axes.

We measure the proximity to a rational cubic motion with singularities using the minimum (over the complete motion cycle) of the sum of squared determinants over the 36 (5×5) sub-matrices of the (6×6) Jacobian matrix composed of the axes' spear coordinates, which yields a univariate polynomial in t. Note that a nonzero value also excludes the possibility that the associated linkage has more than one degree of freedom [HSS13a, Remark 3].

5. Linkage Design

The motion factorization (Sec. 3.2) of the cubic motion designed above (Sec. 4) results in nine cyclic arrangements of six R-joints,

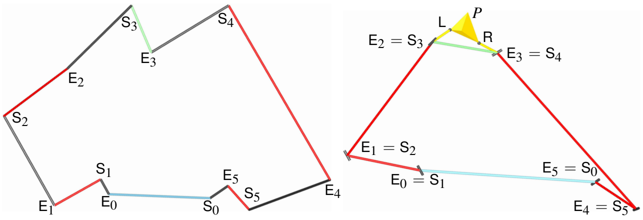


Figure 7: *The base-/end-link is colored in blue/green. (left) Here the links are the common normals of adjacent axes, which is a common concept in robotics. (right) Initialization fulfilling Eq. (11). Moreover, a modification of the end-link caused by the insertion of an end-effector P is illustrated in yellow.*

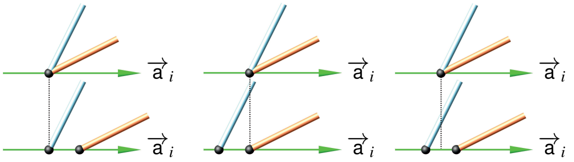


Figure 8: *Two adjacent straight links can be differentiated according to the angle they enclose with the common axis (green). The link implying the smaller/larger angle is colored in orange/blue. The three ways of offsetting labeled by* $+1$ *(left),* -1 *(center) and* 0 *(right), respectively, are illustrated.*

which allow paradoxic mobility. To realize a linkage, these axes have to be connected by links, which can be done in infinitely many ways. We are interested in finding self-collision-free realizations, which is quite a challenging task as even for Bennett mechanisms no theoretical existence results are known.

5.1. Self-Collision-Free Realization

Coros et al. [CTN*13] addressed the collision-avoidance problem, but restricted to the collision-free layering for planar motions. Contrary to [CTN*13], where the problem is tackled algorithmically, a graph-theoretical approach was presented by Qi [Qi19]. Collision-free planar linkages were also constructed for Kempe's Universality Theorem by Abel et al. [ADD*16] and Gallet et al. [GKL*17].

Another approach was proposed by [ZSC16], which inspired the workflow of our collision-avoidance algorithm. Two of its main ingredients are *collision detection* and *segmentation distance* computation [LMK17], which are stressed in the following overview of our algorithm:

After an initial stage of geometric complexity reduction, we apply a particular search strategy for finding a self-collision free linkage, which succeeded in each of the 2000 validation tests (cf. Sec. 6.3). In general, one can perform collision detection numerically when the linkage geometry is established, e.g. [NBA19]. Once a self-collision-free linkage is obtained, we are forced to increase again the complexity of the linkage due to some realization constraints.

Realization Algorithm The detailed procedure consisting of four steps reads as follows:

(1) Initialization: For the initialization of our algorithm we reduce the geometric complexity by assuming that neighboring rotation axes $\overrightarrow{a}_i$ and $\overrightarrow{a}_{i+1}$ (for $i = 0,\ldots,5 \bmod 6$) are linked with straight bars, which are infinitely thin. The starting and end points of these oriented line-segments are denoted with $S_i \in \overrightarrow{a}_{i-1}$ and $E_i \in \overrightarrow{a}_i$ (e.g. Fig. 7, left). A further reduction of the complexity is archived by the assumption that $E_i = S_{i+1}$ (for $i = 0,\ldots,5 \bmod 6$) holds. As a consequence the abstracted linkage consists of a closed polyline $E_0,\ldots,E_5$, whose initial choice results from the following procedure. The user can specify points $E_i \in \overrightarrow{a}_i$, and the remaining ones are computed under the condition

$$\overline{E_0E_1}^2 + \overline{E_1E_2}^2 + \ldots + \overline{E_5E_0}^2 \rightarrow min, \tag{11}$$

(Fig. 7, right) as a short/compact linkage reduces the possibility of a self-collision and for reasons of technical realization. The length of the resulting polyline is denoted by l.

(2) Collision check: We perform collision checks between the six line-segments over the motion cycle. In total, we have to test the following nine pairs of line-segments as neighboring ones share a common vertex: $(0,2)$, $(0,3)$, $(0,4)$, $(1,3)$, $(1,4)$, $(1,5)$, $(2,4)$, $(2,5)$, $(3,5)$. In our case, the collision detection between two links can be reformulated as the determination of intersection points between a line-segment and a ruled surface strip (generated by the relative motion). This problem can be reduced to the computation of real roots of some univariate polynomials of low degree (at most 12), which can quickly detect all possible collisions globally. It is worth mentioning that this strategy is very common in continuous collision detection, e.g. [CWLK06]. If the linkage has collisions, we have to apply the next step.

(3) Search strategy: Zheng et al. [ZSC16] noted that gradient-based methods become inapplicable for optimizing the linkage geometry to avoid collisions due to the discontinuous change of the collision states. In contrast to [ZSC16], we do not use stochastic optimization but apply the following combinatorial search to ensure compactness/shortness of the resulting linkage: We shift the vertices of the initial polyline along the corresponding rotary axes by the distance $\tau\frac{l}{p}$, where p is a user-defined value, τ is a ternary variable taking the values $\{-k, 0, +k\}$ with the iteration index $k \in \mathbb{N}^+$. The user has the possibility to flag a number $f < 6$ of vertices of the initial polyline which are not shifted. The resulting $3^{6-f} - 1$ linkages have to be checked for collision freedom over the motion circle. If none is collision-free, the iteration index k is raised by one, and the procedure is repeated until a solution is obtained.

(4) Link offsetting and thickening: Finally we are forced to increase the complexity of the linkage geometry due to constraints of realization. We have to (i) add a certain thickness to the line segments as they are materialized by cylindrical bars of radius ρ (user-defined value, which depends on material and dimensioning of the linkage) and (ii) separate the points E_i and S_{i+1} along the rotary axis $\overrightarrow{a}_i$ by a distance d of at least 2ρ. The latter offsetting, which allows a simpler and cheaper design of the joints, is done in one of three ways illustrated in Fig. 8. For the resulting 3^6 possible polylines each with 12 vertices, we have to check if two non-adjacent edges have at least distance 2ρ during the complete motion cycle. Due to our offsetting shift, this is guaranteed for the two line-segments S_iE_i and $S_{i+1}E_{i+1}$. Therefore 42 edge-edge distances have to be

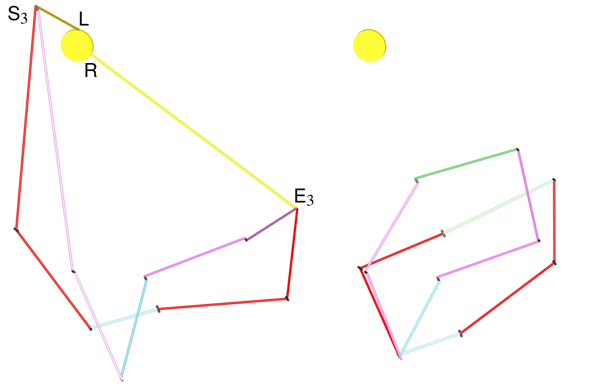


Figure 9: *The initial linkage is illustrated in red and the result obtained by the* realization algorithm *in violet. The base-link is always colored in blue. (left) An end-effector is integrated (yellow disc), where beside* L *and* R *also* S_3 *and* E_3 *are flagged. (right) The result without end-effector integration and flagging of points.*

computed, where each one consists of nine cases (e.g. [Lum85]). If the resulting 42 minimal distances are larger than 2ρ for one of the 3^6 polylines, the algorithm stops; otherwise we go back to step (3) and look for the next collision-free linkage. This final stage of the algorithm requires a minimal distance computation over the motion cycle. The square of the corresponding distance function is a rational function, where the denominator is always positive for any real time t. As a consequence the local extrema of the distance function can again be determined as the real roots of a polynomial.

End-effector Integration in the Realization Algorithm Beside some artistic use of invertible paradoxic 6R-loops, most applications of these structures have to deal with certain functional aspects; e.g. shading in the context of transformable architecture. The user can integrate the so-called end-effector (e.g. shading element) into the *realization algorithm* as follows: We suggest to enclose the end-effector by a polyhedron P with triangular faces, which roughly represents its shape. Moreover, one replaces the end-link S_3E_3 by the composition of P and two line-segments S_3L and RE_3, where L and R are two vertices of P that are flagged not to be updated during the algorithm (Fig. 7, right). If these two points are not specified by the user, they are selected in a way that they are closest to $\vec{a}_2$ and $\vec{a}_3$, respectively. Now the *realization algorithm* works in the same way with the sole difference that also edge-face collision checks and minimal distance computations have to be performed [SJKW02, Vra02].

A comparison of results obtained from the *realization algorithm* with/without integration of an end-effector is illustrated in Fig. 9, which is based on the data of Project 1 presented in Sec. 6.2.

Computational Aspects Note that most parts of the *realization algorithm* can be computed in parallel:

- collision check of all $3^{6-f}-1$ linkages (step (3)), where the nine cases of each check (step 2) can be parallelized in addition,
- 42 minimum distance computations for all 3^6 polylines (step 4), where the nine cases involved in each distance computation can be parallelized additionally.

This also holds if an end-effector is integrated into the *realization algorithm* as the edge-face collision checks and minimal distance computations can be parallelized in an analogous way.

5.2. Generating the Link-Design Spaces

Based on the minimal distances computed in step (4) of the *realization algorithm*, one can easily construct cylindrical design spaces about the line-segments, which are guaranteed to be collision-free; i.e. the user can shape the links inside these cylinders without worrying about collisions. In a post-processing step, one can even try to enlarge the radii of the cylinders by increasing the offsets (not necessarily the same length along all axes) in the linkage obtained from the *realization algorithm* by applying e.g. the Rhino/Grasshopper tool *Galapagos*, which applies evolutionary logic for solving specific optimization problems.

A post-processing algorithm for generating more complicated shaped link-design spaces is described next:

(a) For each link, the user defines a potential link-design space (e.g. a cylinder of rotation around the respective line-segment).
(b) Each potential link-design space is trimmed by the other line-segments of the moving linkage. This process is comparable to wire-cutting.
(c) The boolean difference between each pair of the trimmed link-design spaces is performed over the complete motion cycle. An alternative description of this boolean operation is that the trimmed link-design space is carved out by the swept volume generated by the other involved trimmed link-design space during their relative motion. Note that in contrast to the method of *spacetime geometry carving* proposed by Garg et al. [GJG16] the swept volume approach is not used to avoid collisions, but to create collision-free link-design spaces around the already collision-free linkage.

Note that the order of the pairs of the trimmed link-design spaces and the order within these pairs (as the boolean difference operation is asymmetric) affects the final shape of the resulting link-design spaces (Fig. 10), which are guaranteed collision-free during the complete motion cycle. Therefore the option of ordering gives the user an additional opportunity for interaction in step (c).

Finally it should be mentioned that the boolean operation relies on a voxelisation and is performed by the Rhino/Grasshopper plug-in *Dendro*. Alternatively it can be based on the approximate offset calculation of swept volumes presented in [GJG16].

Loop Grounding The Schatz linkage can be inverted without self-collision, which can easily be verified by moving a model (Fig. 2) of this loop in one's hands. But if one fixes a link of the chain between two finger tips of the left hand and actuates the linkage by the right hand, then the linkage will collide with the left hand during the motion. For sculptural or architectural applications the problem of grounding the loop without restricting its full mobility can also be solved by the above approach (Fig. 10). One only has to take care that the potential link-design space of the base-link is chosen large

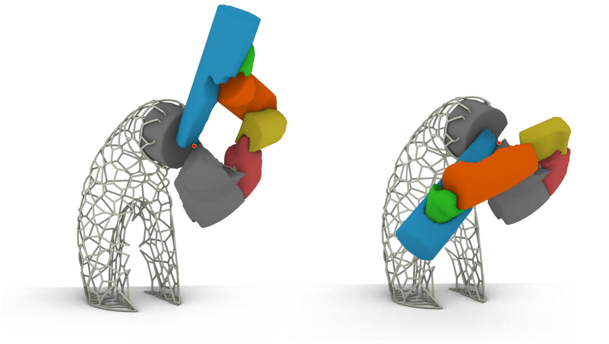


Figure 10: *Illustration of the link-design spaces, where the one for the base-link is colored in gray. The latter is used for solving the loop grounding problem. In general, the grounding of the base-link can be realized by extending one of the adjacent R-joints and fixing it to the ground. The two gray bodies are just connected by this axis (connection point in red).*

enough in step (a). Clearly, the subsequent steps (b) and (c) can carve away this design space, but in all performed examples, the grounding can be achieved over one of the axes situated in the base link (Fig. 10). It is an open problem if this is a general feature of the studied class of linkages.

6. Results & Discussion

6.1. Design & Workflow

Exploiting the design potential offered by invertible paradoxic loop structures is difficult for multiple reasons. With the exception of the invertible cube (Fig. 2), they have never been used, and there is no established design method for the design of these kinematic structures, which are essentially inaccessible by intuitive design approaches. In order to tackle this challenge, we ran a linked master-level studio course and model making class in architecture with twelve students on six design projects. Design experiments led to sample designs that facilitated the development of the presented design tool for invertible paradoxic loop structures.

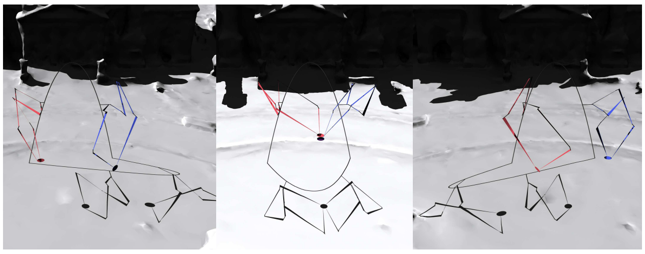


Figure 11: *Visualization of the* Chaotic Relay*: The central figure shows the public square at 1pm on a specific day of the year, where the shadows of the two discs coincide. This circular spot on the floor is shadowed from 10am (left) until 4pm (right). The path of the sun was sampled to obtain a series of target poses for the motion evolution algorithm (Sec. 4.3), where the initial guess was generated by the four pose interpolation (Sec. 4.2).*

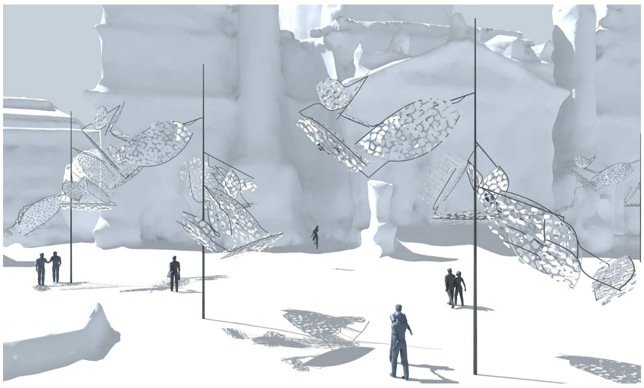


Figure 12: *Visualization of the* Artificial Trees*: Each limb of a tree consists of an invertible paradoxic loop (for details see Fig. 13).*

The students were asked to design a kinetic structure of architectural scale, to be placed in a public square. The task was open to their own interpretation, with focus on sculptural qualities and/or functional aspects of the design, i.e. dynamic shading structures. The students presented their design progress weekly and received feedback from the authors. These feedback sessions also provided valuable information for the authors and showed whether the design tool was suitable, how it was used and which features were missing or needed development. For the more functionally oriented designs, digital simulation served the design evaluation. Updated versions of the design tool were introduced biweekly, and students were taught the underlying mathematical principles.

This approach led to design projects, which are illustrated in Sec. 6.2, and to a useful design tool that is well integrated into existing architectural design workflows. Students could easily design motions, generate, optimize linkages, visualize and analyze their designs in a familiar 3D CAD environment. Complementary physical models were built during the design process to evaluate spatial, performance and design qualities. Extra care and computer-controlled machines were required to achieve the necessary accuracy to ensure the mobility of the models.

6.2. Examples

We present four works that are representative of the range of approaches for all six design projects. Each project posed a specified design problem that was addressed by using our computational tool in conjunction with other CAD and analysis tools.

Project 1, entitled *Chaotic Relay* (Fig. 11), commenced with an analysis of the sun path and exposure in the context of the existing shading pattern at the selected site. This led to the intent to relate the motion of the structure to the motion of the sun. The installation utilizes two linkages to move a shading element in sync with the sun in such a way that its shadow cast on the surface of the public square remains stationary over the course of the day. This simple

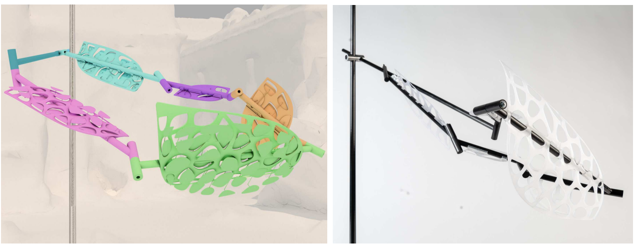


Figure 13: *Visualization (left) and model photo (right) of one limb of an* Artificial Tree *(cf. Fig. 12).*

idea implies a complex design task which is further complicated by the fact that two linkages are utilized to ensure a stationary shadow.

Project 2, entitled *Artificial Trees* (Figs. 12 and 13), also focuses on shading but in a more dynamic manner. The installation relates to the shadow pattern cast by existing trees in the selected site and seeks to mimic this shadow pattern and related light qualities. The treelike kinetic installation, with its arrayed leave-like elements, provides a dynamic shading pattern and improves the sojourn quality of the public space.

Project 3, entitled *Nuloid* (Figs. 1 and 14), focuses on the dynamic sculptural quality of the design and that reflects on the process of the design generation in the actual installation. The shape of the end-link is based on the moving axode of its motion with respect to the base-link, and the other links are shaped by the algorithm described in Sec. 5.2. The project name is a reminiscence to the *oloid*, which is the moving axode of the motion between two opposite links in the invertible cube of Schatz (Fig. 2).

Project 4, entitled *Circadiane* (Fig. 15), is based on the idea to arrange a series of scaled copies of an invertible paradoxic loop. In addition, the linkages are rotated against each other about the common vertical base-link also used for grounding. In this way, the kinetic installation fans out and closes again over one motion cycle synchronized with the day and night period. This dynamic sculpture symbolizes the circadian rhythm of natural phenomena.

Note that animations of all four projects and videos of the two models (Figs. 13 and 14) are uploaded as supplementary material, which also includes a video of a Schatz linkage and screen captures of the developed Rhino/Grasshopper plug-in during a user session.

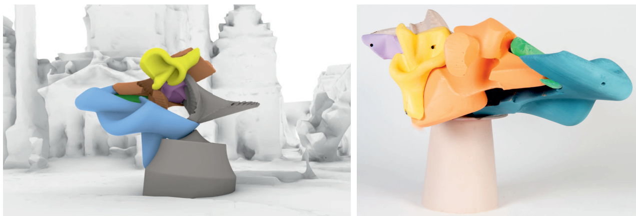


Figure 14: *Visualization of the* Nuloid *(left): The six links of the invertible paradoxic loop have different colors, where the base-link (including the grounding structure) is gray, and the end-link is orange. Model photo of the* Nuloid *(right).*

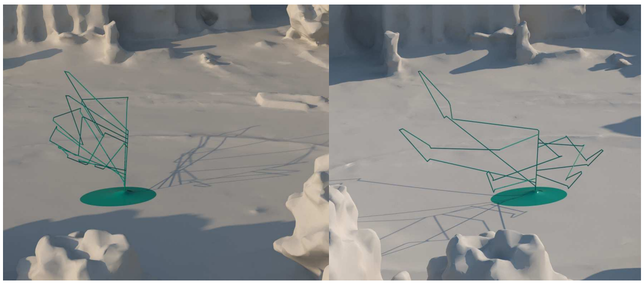


Figure 15: *Visualization of the* Circadiane*: Closed configuration in the morning (left) and the fanned out one in the evening (right).*

6.3. Validation

In a first step, the conceptual design of the *realization algorithm* (Sec. 5.1) as well as the *evolution algorithm* (Sec. 4.3) were developed based on *Maple18* implementations, which were also used to check the correctness of the later *C#* Rhino/Grasshopper plug-in. In the following, we provide a statistic validation of these two algorithms.

Statistic Validation of the Realization Algorithm We randomly generated 1000 cubic rational motions, factorized them and combined the resulting factorizations to obtain *angle-symmetric 6R loops* as well as *double Bennett 6R loops*. The offset distance d of step (4) is set to $1/100$ of the length of the linkage resulting from step (3), which depends on the parameter p, chosen as 10. Choosing $\rho = d/4$, we produced Table 1 with *Maple18*, where the following mean values are given:

- $\varnothing_k$ of the iteration index k of step (3).
- $\varnothing_c$ of collision-checks of polylines with 6 vertices (step (3)).
- $\varnothing_b$ of the number of steps backwards from (4) to (3).
- $\varnothing_m$ of minimal distance computations for polylines with 12 vertices (step (4)).
- $\varnothing_r$ of the ratio between the length of the final polylines with 12 vertices and l.

Moreover, we counted the number z of successful minimal distance computations resulting from the offsetting sequence $\{0,0,0,0,0,0\}$, which was selected to be the first one in the list of these 3^6 sequences. If this offsetting sequence failed, we used as next sequence the one implied by the sign function of the shift sequence of step (3); e.g. the shift sequence $\{+k,+k,0,-k,0,-k\}$ yields the offsetting sequence $\{+1,+1,0,-1,0,-1\}$. In how many cases this sequence succeeded is given by the number s.

Table 1: *Statistic validation of the* realization algorithm *(DB denotes the double Bennett 6R loop and AS the angle-symmetric one).*

	$\varnothing_k$	$\varnothing_c$	$\varnothing_b$	$\varnothing_m$	$\varnothing_r$	z	s
AS	1.355	1169.782	.687	555.284	1.416	506	88
DB	1.100	928.379	.764	613.044	1.346	590	55

Statistic Validation of the Evolution Algorithm We ran our evolution algorithm in *Maple18* for a single sequence of ten ordered target poses with 500 different initial guesses, which were generated randomly according to Eq. (9). The target poses were sampled from a random cubic rational motion, thus our algorithm can be evaluated by the sum of the squared distances (8) of the target poses to the closest poses of the resulting cubic motion. This value is indicated by the vertical axis of the graph displayed in Fig. 16 and its horizontal axis gives the number of iterations. Moreover, it should be noted that the object dependent metric was based on an inertia sphere of radius 1 and that the trajectory length of the barycenter under the targeted cubic motion equals 20.

One can see that the cost-function in terms of the metric is decreasing, where the mean value is plotted in green. The red curve indicates the best approximation within 34 steps and the blue curve displays the worst of the 500 runs.

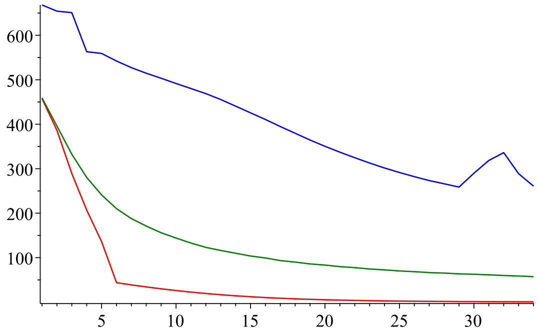


Figure 16: *Statistic validation of the* evolution algorithm.

It can happen that the *evolution algorithm* produces a motion $\underline{\mathfrak{C}}^e(t)$ that makes an unwanted loop in between two adjacent target poses (Fig. 17). Such loops cannot be straightened using the guiding pose approach (2a) during the middle and final stage. In this case, one can either try another random initial guess (9) or the following modified procedure (Fig. 17):

- Insert virtual target poses between adjacent target poses based on the motion interpolation given in [PHR04, HP04], as it takes the metric (8) into account.
- Compute in every stage the guiding poses with approach (2b).
- Increase the number of iteration steps by bounding the step size.

Validation through the Design Projects The role of the design projects in terms of validation included the identification of problems in the different versions of the tool, which were revealed through applications. In this context, the openness of the design brief ensured a breadth of design approaches resulting in a broader range of testing of the intended range of functionality and applicability of the tool. In a projected next phase of the tool development, it would be useful to define the design brief in more narrow terms in relation to specifically intended functionality of the tool so as to develop the latter in greater detail.

6.4. Limitations & Future Work

If we assume that one R-joint is actuated with constant velocity (e.g. 10s for one rotation), then it can happen that another R-joint of the linkage rotates extremely fast in a short period (e.g. 300 deg within $2s$) and in the remaining time very slowly (e.g. 60 deg within $8s$). Beside the large velocities, the resulting big accelerations are problematic for e.g. architectural applications. Hence in a future work, physical aspects (inertia forces, wind loads, etc.) should be considered. Further topics dedicated to future research are e.g. the:

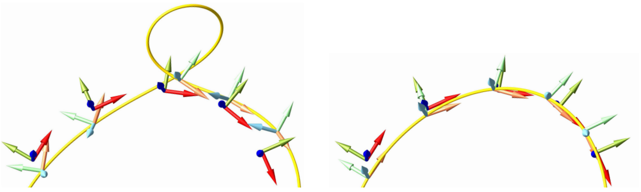


Figure 17: *The loop phenomenon (left) can be avoided by a more costly modification of the evolution algorithm (right).*

- modification of our design tool for Bennett mechanisms and Goldberg linkages (tracing a special rational cubic motion according to Hegedüs et al. [HSS13b]), which is straightforward.
- extension of our design tool to paradoxic loop structures with a prismatic (P) joint (e.g. P5R or P4R linkages). For practical applications, only one actuated P-joint makes sense as passive P-joints can easily jam.
- development of a computational tool for the design of networks of rational 6R linkages. Networks for the special case of type III Bricard linkages are discussed in [LZD*19, Bak09].
- classification of all 6R loops with a rational mobility possessing singular configurations. Based on this theoretical result, such linkages can actively be avoided during the design process.
- performance comparison of the *evolution algorithm* based on the l_2-norm (Sec. 4.3) and the l_1-norm [FH10].

Finally it should be pointed out that based on the experience gained during the presented research project, design experiments should play again a central role in the further development of the tool as they stimulate a fruitful interplay between kinematic geometry and architecture/design.

Acknowledgments

The research was funded by the TOP-program project "*Geometry and Computational Design for Architecture and Fabrication*" of TU Wien. The research was funded by the Austrian Science Fund (FWF): P 24927-N25 and P 31061. This research was partially supported by the Austrian Science Fund (FWF): W1214-N15, project DK9. We thank Helmut Pottmann, Hans-Peter Schröcker and Martin Kilian for illuminating discussions. Special thanks to the following students for preparing the presented projects: Georg Holzmann (Project 1), Michaela Gebetsroither & Michaela Nömayr (Project 2), Asrin Palantöken & Marius Valente (Project 3) and Georg Lobe (Project 4).

References

[ADD*16] Abel Z., Demaine E. D., Demaine M. L., Eisenstat S., Lynch J., Schardl T. B.: Who needs crossings? Hardness of plane graph rigidity. In *32nd International Symposium on Computational*

Geometry (SoCG 2016) (2016), Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 8

[AJ07] Aigner M., Jüttler B.: Hybrid curve fitting. *Computing 79*, 2-4 (2007), 237–247. 6

[Bak09] Baker E. J.: On the skew network corresponding to Bricard's doubly collapsible octahedron. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 223*, 5 (2009), 1213–1221. 7, 12

[Ben03] Bennett G. T.: A new mechanism. *Engineering 76* (1903), 777–778. 3

[Bla60] Blaschke W.: *Kinematik und Quaternionen*, vol. 4 of *Mathematische Monographien*. VEB Deutscher Verlag der Wissenschaften, Berlin, Germany, 1960. 4

[BLZM16] Barozzi M., Lienhard J., Zanelli A., Monticelli C.: The sustainability of adaptive envelopes: developments of kinetic architecture. *Procedia Engineering 155* (2016), 275–284. 2

[BR79] Bottema O., Roth B.: *Theoretical Kinematics*, vol. 24 of *Applied Mathematics and Mechanics*. North-Holland Publishing Company, Amsterdam, Netherlands, 1979. 4

[BSH05] Brunnthaler K., Schröcker H.-P., Husty M.: A new method for the synthesis of Bennett mechanisms. In *International Workshop on Computational Kinematics (CK2005), Cassino, Italy, May* (2005), pp. 4–6. 3, 5

[BSK*13] Barton M., Shi L., Kilian M., Wallner J., Pottmann H.: Circular arc snakes and kinematic surface generation. *Comput. Graph. Forum 32*, 2 (2013), 1–10. 2

[Cal81] Calatrava S.: *Zur Faltbarkeit von Fachwerken*. PhD thesis, Eidgenössische Technische Hochschule Zürich ETH, Zürich, Switzerland, 1981. 3

[CAL17] Chernousko F. L., Akulenko L. D., Leshchenko D. D.: *Evolution of motions of a rigid body about its center of mass*. Springer, 2017. 6

[CDIO04] Cantarella J. H., Demaine E. D., Iben H. N., O'Brien J. F.: An energy-driven approach to linkage unfolding. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry* (New York, NY, USA, 2004), SCG '04, ACM, pp. 134–143. 3

[Cli71] Clifford W. K.: Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society s1-4*, 1 (1871), 381–395. 4

[CTN*13] Coros S., Thomaszewski B., Noris G., Sueda S., Forberg M., Sumner R. W., Matusik W., Bickel B.: Computational design of mechanical characters. *ACM Trans. Graph. 32*, 4 (2013), 83:1–83:12. 2, 8

[CWLK06] Choi Y.-K., Wang W., Liu Y., Kim M.-S.: Continuous collision detection for two moving elliptic disks. *IEEE Transactions on Robotics 22*, 2 (2006), 213–224. 8

[CY12] Chen Y., You Z.: Spatial Overconstrained Linkages - The Lost Jade. In *Explorations in the History of Machines and Mechanisms*, Koetsier T., Ceccarelli M., (Eds.), vol. 15 of *History of Mechanism and Machine Science*. Springer Netherlands, 2012, pp. 535–550. 2, 3

[Die95] Dietmaier P.: *Einfach übergeschlossene Mechanismen mit Drehgelenken*. Habilitation thesis, Graz University of Technology, Graz, Austria, 1995. 2, 3, 5

[Dim48] Dimentberg F. M.: A general method for the investigation of finite displacements of spatial mechanisms and certain cases of passive joints. *Trudi Seminara po Teorii Mashin i Mekhanizmov 5*, 17 (1948), 5–39. 4

[Esc13] Escrig F.: Modular, light and transformable architecture. In *New Proposals for Transformable architecture. Engineering and Design in the honor of Emilio Perez Pinero* (Seville, Spain, 2013), Starbooks, pp. 11–22. 3

[EWMH14] Elghazi Y., Wagdy A., Mohamed S., Hassan A.: Daylighting driven design: Optimizing kaleidocycle facade for hot arid climate. In *5th German-Austrian Conference of the International Building Performance Simulation Association (BauSIM 2014): Human-centred building(s)* (Aachen, Germany, 2014), van Treeck C., Müller D., (Eds.), International Building Performance Simulation Association, pp. 314–321. 3

[FH10] Flöry S., Hofer M.: Surface fitting and registration of point clouds using approximations of the unsigned distance function. *Computer Aided Geometric Design 27*, 1 (2010), 60–77. 12

[FT13] Felipe S., Truco J.: Hypermembrane Demo DHUB, 2013. URL: https://www.hybridarch.net/hypermembrane-demo. 2

[Gai15] Gaitan J.: *POP UP – A deployable brand in the urban fabric*. Master's thesis, Carleton University, Ottawa, Canada, 2015. 3

[GJG16] Garg A., Jacobson A., Grinspun E.: Computational design of reconfigurables. *ACM Trans. Graph. 35*, 4 (2016), 90:1–90:14. 2, 9

[GKL*17] Gallet M., Koutschan C., Li Z., Regensburger G., Schicho J., Villamizar N.: Planar linkages following a prescribed motion. *Mathematics of Computation 86*, 303 (2017), 473–506. 8, 15

[Gol43] Goldberg M.: New five-bar and six-bar linkages in three dimensions. *Trans. ASME 65* (1943), 649–656. 3

[GR94] Ge Q. J., Ravani B.: Computer aided geometric design of motion interpolants. *ASME Journal of Mechanical Design 116*, 3 (1994), 756–762. 4

[HAI90] Hoberman Associates, Inc., 1990. URL: http://www.hoberman.com/. 2, 3

[Ham11] Hamann M.: Line-symmetric motions with respect to reguli. *Mechanism and Machine Theory 46*, 7 (2011), 960–974. 5

[Han12] Hanson A. J.: Quaternion applications. In *SIGGRAPH Asia 2012 Courses* (New York, NY, USA, 2012), SA '12, ACM, pp. 11:1–11:20. 4

[HP04] Hofer M., Pottmann H.: Energy-minimizing splines in manifolds. *ACM Trans. Graph. 23*, 3 (2004), 284–293. 12

[HSS13a] Hegedüs G., Schicho J., Schröcker H.-P.: Factorization of rational curves in the Study quadric. *Mechanism and Machine Theory 69* (2013), 142–152. 3, 4, 5, 7, 15

[HSS13b] Hegedüs G., Schicho J., Schröcker H.-P.: The theory of bonds: A new method for the analysis of linkages. *Mechanism and Machine Theory 70*, 0 (2013), 407–424. 12

[HSS15] Hegedüs G., Schicho J., Schröcker H.-P.: Four-pose synthesis of angle-symmetric 6R linkages. *Journal of Mechanisms and Robotics 7*, 4 (2015), 041006. 6

[Jüt93] Jüttler B.: Über zwangläufige rationale Bewegungsvorgänge. *Sitzungsberichte der Mathematisch-Naturwissenschaftlichen Klasse, Abteilung II 202*, 1-10 (1993), 117–232. 4

[JW02] Jüttler B., Wagner M. G.: Kinematics and animation. In *Handbook of Computer Aided Geometric Design*, Farin G., Hoschek J., Kim M.-S., (Eds.). North-Holland, Amsterdam, 2002, pp. 723–748. 4

[Kar98] Karger A.: Classification of 5R closed kinematic chains with self mobility. *Mechanism and Machine Theory 33*, 1 (1998), 213–222. 3

[KCvO08] Kavan L., Collins S., Žára J., O'Sullivan C.: Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (2008), 105:1–105:23. 4

[KKP19] Kaji S., Kajiwara K., Park H.: *Linkage Mechanisms Governed by Integrable Deformations of Discrete Space Curves*. Tech. Rep. arXiv:1903.06360 [nlin.SI], arXiv, 2019. 3

[KLPCP18] Konaković-Luković M., Panetta J., Crane K., Pauly M.: Rapid deployment of curved surfaces via programmable auxetics. *ACM Trans. Graph. 37*, 4 (2018), 106:1–106:13. 2

[KMM17] Kilian M., Monszpart A., Mitra N. J.: String actuated curved folded surfaces. *ACM Trans. Graph. 36*, 3-4 (2017). 2

[Kot95] KOTELNIKOV A. P.: Screw calculus and some applications to geometry and mechanics. *Annals of the Imperial University of Kazan 62* (1895). 4

[KR92] KAZEROUNIAN K., RASTEGAR J.: Object norms: A class of coordinate and metric independent norms for displacements. In *Flexible Mechanisms, Dynamics and Analysis* (1992), Kinzel G. L., (Ed.), American Society of Mechanical Engineers, pp. 271–275. 5

[LH16] LE B. H., HODGINS J. K.: Real-time skeletal skinning with optimized centers of rotation. *ACM Trans. Graph. 35*, 4 (2016), 37:1–37:10. 4

[Li15] LI Z.: *Closed Linkages with Six Revolute Joints*. PhD thesis, Johann Radon Institute for Computational and Applied Mathematics, Linz, Austria, 2015. 2, 3

[LLF80] Len Lye Foundation, 1980. URL: http://www.lenlyefoundation.com/. 3

[LMK17] LIN M. C., MANOCHA D., KIM Y. J.: Collision and proximity queries. In *Handbook of Discrete and Computational Geometry*, Goodman J. E., O'Rourke J., Tóth C. D., (Eds.), 3 ed. CRC Press, Inc., Boca Raton, FL, USA, 2017, ch. 39, pp. 1029–1056. 8

[LS13] LI Z., SCHICHO J.: Classification of angle-symmetric 6R linkages. *Mechanism and Machine Theory 70*, 0 (2013), 372–379. 5

[LSS16a] LI Z., SCHICHO J., SCHRÖCKER H.-P.: The rational motion of minimal dual quaternion degree with prescribed trajectory. *Computer Aided Geometric Design 41* (2016), 1–9. 4

[LSS16b] LI Z., SCHICHO J., SCHRÖCKER H.-P.: Spatial straight-line linkages by factorization of motion polynomials. *ASME Journal of Mechanisms and Robotics 8*, 2 (2016), 021002. 3

[LSS18] LI Z., SCHICHO J., SCHRÖCKER H.-P.: Kempe's universality theorem for rational space curves. *Foundations of Computation Mathematics 18* (2018), 509–536. 3, 5, 15

[Lum85] LUMELSKY V. J.: On fast computation of distance between line segments. *Information Processing Letters 21*, 2 (1985), 55–61. 9

[LZD*19] LU S., ZLATANOV D., DING X., ZOPPI M., GUEST S. D.: A network of type III Bricard linkages. *ASME Journal of Mechanisms and Robotics 11*, 1 (2019), 011013. 12

[Mar16] MARYSSE C.: *Structural Adaptive Facades*. Master's thesis, Ghent University, Belgium, 2016. 2

[MS11] MCCARTHY M. J., SOH G. S.: *Geometric Design of Linkages*, 2nd edition ed., vol. 11 of *Interdisciplinary Applied Mathematics*. Springer, Heidelberg, Germany, 2011. 2, 3

[MSZ94] MURRAY R. M., SASTRY S. S., ZEXIANG L.: *A Mathematical Introduction to Robotic Manipulation*, 1st ed. CRC Press, Inc., Boca Raton, FL, USA, 1994. 5

[MZB*17] MEGARO V., ZEHNDER J., BÄCHER M., COROS S., GROSS M., THOMASZEWSKI B.: A computational design tool for compliant mechanisms. *ACM Trans. Graph. 36*, 4 (2017), 82:1–82:12. 2

[Naw19] NAWRATIL G.: Kinematic interpretation of the Study quadric's ambient space. In *Advances in Robot Kinematics 2018* (2019), Lenarcic J., Parenti-Castelli V., (Eds.), Springer, pp. 3–11. 4

[NBA19] NISHIDA G., BOUSSEAU A., ALIAGA D.: Multi-pose interactive linkage design. *Comput. Graph. Forum 38*, 2 (2019). 2, 8

[OWHB17] OVERVELDE J. T. B., WEAVER J. C., HOBERMAN C., BERTOLDI K.: Rational design of reconfigurable prismatic architected materials. *Nature 541* (2017), 347–352. 2

[Par95] PARK F. C.: Distance metrics on the rigid-body motions with applications to mechanism design. *ASME Journal of Mechanical Design 117*, 1 (1995), 48–54. 5

[PEVW15] POTTMANN H., EIGENSATZ M., VAXMAN A., WALLNER J.: Architectural geometry. *Computers and Graphics 47* (2015), 145–164. 2

[PG05] PURWAR A., GE Q. J.: On the effect of dual weights in computer aided design of rational motions. *ASME Journal of Mechanical Design 127*, 5 (2005), 967–972. 4

[PHR04] POTTMANN H., HOFER M., RAVANI B.: Variational motion design. In *On Advances in Robot Kinematics* (2004), Lenarcic J., Galetti C., (Eds.), Springer, pp. 361–370. 5, 12

[Pic06] PICKOVER C. A.: *The Möbius Strip: Dr. August Möbius's Marvelous Band in Mathematics, Games, Literature, Art, Technology, and Cosmology*. Thunder's Mouth Press, New York, USA, 2006. 3

[PKLI*19] PANETTA J., KONAKOVIĆ-LUKOVIĆ M., ISVORANU F., BOULEAU E., PAULY M.: X-shells: A new class of deployable beam structures. *ACM Trans. Graph. 38*, 4 (2019), 83:1–83:15. 2

[POT17] PÉREZ J., OTADUY M. A., THOMASZEWSKI B.: Computational design and automated fabrication of Kirchhoff-Plateau surfaces. *ACM Trans. Graph. 36*, 4 (2017), 62:1–62:12. 2

[PSH18] PFURNER M., SCHRÖCKER H.-P., HUSTY M.: Path planning in kinematic image space without the Study condition. In *Advances in Robot Kinematics 2016* (2018), Lenarcic J., Merlet J.-P., (Eds.), Springer, pp. 285–292. 4

[PTV*17] PIETRONI N., TARINI M., VAXMAN A., PANOZZO D., CIGNONI P.: Position-based tensegrity design. *ACM Trans. Graph. 36*, 6 (2017), 172:1–172:14. 2

[PW11] POTTMANN H., WALLNER J.: *Computational Line Geometry*. Springer, Berlin, Germany, 2011. 7

[Qi19] QI J.: *How to avoid collision of 3D-realization for moving graphs*. Tech. Rep. arXiv:1904.13260 [math.CO], arXiv, 2019. 8

[Rad18] RAD T.-D.: *Factorization of Motion Polynomials and its Application in Mechanism Science*. PhD thesis, University of Innsbruck, Innsbruck, Austria, 2018. 6, 7

[RHSH18] RABINOVICH M., HOFFMANN T., SORKINE-HORNUNG O.: Discrete geodesic nets for modeling developable surfaces. *ACM Trans. Graph. 37*, 2 (2018), 16:1–16:17. 2

[Rös98] RÖSCHEL O.: Rational motion design – a survey. *Computer-Aided Design 30*, 3 (1998), 169–178. Motion Design and Kinematics. 4

[RS18] RAD T.-D., SCHRÖCKER H.-P.: Optimal synthesis of overconstrained 6R linkages by curve evolution. In *Computational Kinematics*. Springer, 2018, pp. 535–543. 3, 6, 7

[Sar53] SARRUS P. F.: Note sur la transformation des mouvements rectilignes alternatifs, en mouvements circulaires: et rèciproquement. *Comptes Rendus des Séances de l'Académie des Sciences de Paris 36* (1853), 1036–1038. 2, 3

[sch04] SCHNEIDER+SCHUMACHER: Blossom hotel kish island, 2004. URL: https://www.schneider-schumacher.com/projects/. 3

[Sch13] SCHATZ P.: *The Study of Rhythms and Technology: The Evertible Cube, Polysomatic Form-Finding*. Niggli, Sulgen, Switzerland, 2013. English reprint of the book "Rhythmusforschung und Technik" from 1975. 1

[SF19] SCHÖNKE J., FRIED E.: Single degree of freedom everting ring linkages with nonorientable topology. *Proceedings of the National Academy of Sciences 116*, 1 (2019), 90–95. 3

[SJKW02] SOHN K.-A., JÜTTLER B., KIM M.-S., WANG W.: Computing distances between surfaces using line geometry. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.* (2002), IEEE, pp. 236–245. 9

[SLRP18] SHANG X., LIU L., RAFSANJANI A., PASINI D.: Durable bistable auxetics made of rigid solids. *Journal of Materials Research 33*, 3 (2018), 300–308. 2

[ST19] SARABANDI S., THOMAS F.: A Survey on the Computation of Quaternions From Rotation Matrices. *Journal of Mechanisms and Robotics 11*, 2 (2019). 021006. 15

[Stu91] STUDY E.: Von den Bewegungen und Umlegungen. *Mathematische Annalen 39*, 4 (1891), 441–565. 4

[SW77] Schattschneider D., Walker W.: *M. C. Escher Kaleidocycles*. Ballantine, New York, USA, 1977. 3

[SWC18] Selig J. M., Wu Y., Carricato M.: Motion interpolation in Lie subgroups and symmetric subspaces. In *Computational Kinematics* (2018), Zeghloul S., Romdhane L., Laribi M. A., (Eds.), Springer, pp. 467–474. 4

[Tac10] Tachi T.: Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. In *Advances in Architectural Geometry 2010* (Vienna, 2010), Ceccato C., Hesselgren L., Pauly M., Pottmann H., Wallner J., (Eds.), Springer Vienna, pp. 87–102. 2

[TCG*14] Thomaszewski B., Coros S., Gauge D., Megaro V., Grinspun E., Gross M.: Computational design of linkage-based characters. *ACM Trans. Graph. 33*, 4 (2014), 64:1–64:9. 2

[TH17] Tachi T., Hull T. C.: Self-foldability of rigid origami. *ASME Journal of Mechanisms and Robotics 9*, 2 (2017), 021008. 2

[TJS90] Theo Jansen's Strandbeest, 1990. URL: https://www.strandbeest.com/. 3

[VG13] Viquerat A. D., Guest S. D.: Designing folding rings using polynomial continuation. *ASME Journal of Mechanisms and Robotics 6*, 1 (2013), 011005. 3

[Vra02] Vranek D.: Fast and accurate circle-circle and circle-line 3D distance computation. *Journal of graphics tools 7*, 1 (2002), 23–31. 9

[Wal67] Walker W.: Foldable structure, 1967. U.S. Patent No. 3302321 , February 7th, 1967. 3

[Yan63] Yang A. T.: *Application of quaternion algebra and dual numbers to the analysis of spatial mechanisms*. PhD thesis, Columbia University, New York, USA, 1963. 4

[YCC17] Yu C., Crane K., Coros S.: Computational design of telescoping structures. *ACM Trans. Graph. 36*, 4 (2017), 83:1–83:9. 2

[YYL*19] Yu M., Ye Z., Liu Y.-J., He Y., Wang C. C. L.: Lineup: Computing chain-based physical transformation. *ACM Trans. Graph. 38*, 1 (2019), 11:1–11:16. 3

[YZC18] Yuan Y., Zheng C., Coros S.: Computational design of transformables. *Comput. Graph. Forum* (2018). 2

[ZAC*17] Zhang R., Auzinger T., Ceylan D., Li W., Bickel B.: Functionality-aware retargeting of mechanisms to 3D shapes. *ACM Trans. Graph. 36*, 4 (2017), 81:1–81:13. 2

[ZSC16] Zheng C., Sun T., Chen X.: Deployable 3D linkages with collision avoidance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, Germany, 2016), SCA '16, Eurographics Association, pp. 179–188. 2, 3, 8

Appendix A: Spatial displacement in terms of unit dual quaternions

Usually a spatial displacement is given in the form $\mathbf{x} \mapsto \mathbf{y} := \mathbf{R}\mathbf{x} + \mathbf{t}$, where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ the translation vector. The pair $(\mathbf{R}, \mathbf{t})$ encodes an element of SE(3) that can also be represented by a unit dual quaternion $\underline{\mathfrak{Q}} = \mathfrak{P} + \varepsilon\mathfrak{D}$ which can be computed according to [ST19].

By embedding the points of $\mathbb{R}^3$ into the set of unit dual quaternions by

$$\mathbf{x} = (x_1, x_2, x_3) \mapsto \underline{\mathfrak{X}} := 1 + \varepsilon\mathfrak{X} \quad \text{with} \quad \mathfrak{X} := x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \tag{12}$$

the spatial displacement can be written in terms of unit dual quaternions by

$$\underline{\mathfrak{X}} \mapsto \underline{\mathfrak{Y}} := (\mathfrak{P} + \varepsilon\mathfrak{D})\underline{\mathfrak{X}}(\widetilde{\mathfrak{P}} - \varepsilon\widetilde{\mathfrak{D}}). \tag{13}$$

This implies

$$\mathfrak{X} \mapsto \mathfrak{Y} := \mathfrak{P}\mathfrak{X}\widetilde{\mathfrak{P}} + (\mathfrak{D}\widetilde{\mathfrak{P}} - \mathfrak{P}\widetilde{\mathfrak{D}}). \tag{14}$$

Appendix B: Bennett flip

The *Bennett flip* is a factorization trick for quadratic motion polynomial (see [GKL*17] for planar and [LSS18] for spatial motions).

Definition 1 The *Bennett flip* is the map

$$\text{bflip}\colon \mathbb{DH}^2 \setminus \{(\underline{\mathfrak{A}}_1, \underline{\mathfrak{A}}_2) \mid \widetilde{\underline{\mathfrak{A}}}_1 = \underline{\mathfrak{A}}_2\} \to \mathbb{DH}^2,\ (\underline{\mathfrak{A}}_1, \underline{\mathfrak{A}}_2) \mapsto (\underline{\mathfrak{A}}'_1, \underline{\mathfrak{A}}'_2)$$

with $\underline{\mathfrak{A}}'_2 = -(\widetilde{\underline{\mathfrak{A}}}_1 - \underline{\mathfrak{A}}_2)^{-1}(\underline{\mathfrak{A}}_1\underline{\mathfrak{A}}_2 - \underline{\mathfrak{A}}_1\widetilde{\underline{\mathfrak{A}}}_1)$ and $\underline{\mathfrak{A}}'_1 = \underline{\mathfrak{A}}_1 + \underline{\mathfrak{A}}_2 - \underline{\mathfrak{A}}'_2$.

The interpretation in terms of factorizations of motion polynomial is $(t - \underline{\mathfrak{A}}_1)(t - \underline{\mathfrak{A}}_2) = (t - \underline{\mathfrak{A}}'_1)(t - \underline{\mathfrak{A}}'_2)$. In general the axes $\vec{a}_1$, $\vec{a}_2$, $\vec{a}'_2$, $\vec{a}'_1$ form a Bennett linkage and the following relations hold for their norm polynomials:

$$\begin{aligned} (t - \underline{\mathfrak{A}}_1)(t - \widetilde{\underline{\mathfrak{A}}}_1) &= (t - \underline{\mathfrak{A}}'_2)(t - \widetilde{\underline{\mathfrak{A}}}'_2) \\ (t - \underline{\mathfrak{A}}_2)(t - \widetilde{\underline{\mathfrak{A}}}_2) &= (t - \underline{\mathfrak{A}}'_1)(t - \widetilde{\underline{\mathfrak{A}}}'_1). \end{aligned} \tag{15}$$

We can obtain another factorization just by applying *Bennett flips* continually.

Theorem 1 Given a generic monic motion polynomial $\underline{\mathfrak{M}}(t)$ of degree n with a known factorization

$$\underline{\mathfrak{M}}(t) = (t - \underline{\mathfrak{A}}_0) \cdots (t - \underline{\mathfrak{A}}_{n-1}), \tag{16}$$

another factorization can be obtained by recursively applying Bennett flips to two neighboring factors.

Proof Let us apply a Bennett flip to two neighboring linear factors of the factorization (16), for instance, for some $0 \le s < n-1$,

$$\begin{aligned} \underline{\mathfrak{M}}(t) &= (t - \underline{\mathfrak{A}}_0) \cdots (t - \underline{\mathfrak{A}}_s)(t - \underline{\mathfrak{A}}_{s+1}) \cdots (t - \underline{\mathfrak{A}}_{n-1}) \\ &= (t - \underline{\mathfrak{A}}_0) \cdots (t - \underline{\mathfrak{A}}'_s)(t - \underline{\mathfrak{A}}'_{s+1}) \cdots (t - \underline{\mathfrak{A}}_{n-1}). \end{aligned} \tag{17}$$

By the factorization algorithm in [HSS13a], a factorization

$$\underline{\mathfrak{M}}(t) = (t - \underline{\mathfrak{A}}_0) \cdots (t - \underline{\mathfrak{A}}_s)(t - \underline{\mathfrak{A}}_{s+1}) \cdots (t - \underline{\mathfrak{A}}_{n-1})$$

depends on an ordering of the n quadratic factors of the norm polynomial

$$\underline{\mathfrak{M}}(t)\widetilde{\underline{\mathfrak{M}}}(t) = P_0 P_1 \cdots P_{n-1},$$

where $P_s = (t - \underline{\mathfrak{A}}_s)(t - \widetilde{\underline{\mathfrak{A}}}_s)$ for $s = 0, \dots, n-1$. Hence, two factorizations (if they exist) are different if and only if their two corresponding factor orderings are different. By the equation (15), we have two corresponding factor orderings

$$\begin{aligned} \underline{\mathfrak{M}}(t)\widetilde{\underline{\mathfrak{M}}}(t) &= P_0 P_1 \cdots P_s P_{s+1} \cdots P_{n-1} \\ &= P_0 P_1 \cdots P_{s+1} P_s \cdots P_{n-1}. \end{aligned} \tag{18}$$

If the two norm polynomials P_s and P_{s+1} are not equal, then we have another different factorization. Therefore, applying one Bennett flip, we can at most swap the order of one pair of two neighboring quadratic factors. One might need several flips for obtaining a wanted factorization from a known one. □