Invertible Paradoxic Loop Structures for Transformable Design

Zijia Li

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University Linz, Austria

May 25-29, 2020

Joint work with G. Nawratil, F. Rist and M. Hensel

Center for Geometry and Computational Design, TU Wien, Austria

Contents

Our aim

Mechanism

- aesthetic transformation under functional aspects (e.g. shading)
- 1-parameteric mobility (time t of the day)
- invertible loop (no need for returning in the night)
- rational motion for synthesis

Our aim

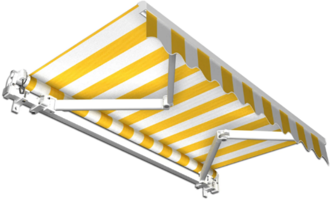
Mechanism

- aesthetic transformation under functional aspects (e.g. shading)
- 1-parameteric mobility (time t of the day)
- invertible loop (no need for returning in the night)
- rational motion for synthesis

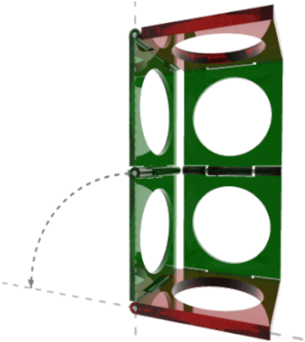
Design focusing on architectural and artistic application

- An easy-to-use package (Rhino/Grasshopper plug-in)
- Interaction

Transformable loop



(a) Awning



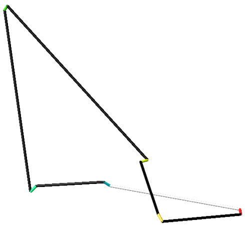
(b) Sarrus linkage

- Only straight-line motion

Ivertible cube

(a) Schatz invertible cube

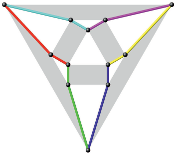
Our invertible loop



Sketching the workflow

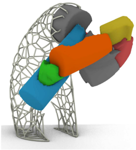
Motion Design

⟶



Motion Factorization

⟶



Linkage Design

Basic notations

$\mathrm{SE}(3)$

- **Special Euclidean group** $\mathrm{SE}(3)(\mathbb{R})$ is defined as the group of all maps from $\mathbb{R}^3$ to itself preserving distance and orientation.

Basic notations

$\mathrm{SE}(3)$

- **Special Euclidean group** $\mathrm{SE}(3)(\mathbb{R})$ is defined as the group of all maps from $\mathbb{R}^3$ to itself preserving distance and orientation.

$\mathbb{DH}$

- $\mathbb{DH}$ (**dual quaternions**): 8-dimensional real vector space generated by $1, \mathbf{i}, \mathbf{j}, \mathbf{k}, \epsilon, \epsilon\mathbf{i}, \epsilon\mathbf{j}, \epsilon\mathbf{k}$, where $\epsilon^2 = 0$.
- **Study quadric** $\Psi = \{\underline{\mathfrak{h}} \in \mathbb{DH} | \ \underline{\mathfrak{h}} \ \underline{\widetilde{\mathfrak{h}}} \in \mathbb{R}\}$ and $G = \{\underline{\mathfrak{h}} \in S | \ \underline{\mathfrak{h}} \ \underline{\widetilde{\mathfrak{h}}} = 0\}$.
- The complement $\Psi - G$ can be identified with $\mathrm{SE}(3)$ by an isomorphism:

$$\alpha : (\Psi - G)/\mathbb{R}^* \to \mathrm{SE}(3)$$

Back projection

Points of the ambient space can be projected onto the Study quadric with a mapping

$$\phi : \mathbb{P}^7 \setminus G \to \Psi \setminus G$$

maps a dual quaternion $\mathfrak{P} + \epsilon\mathfrak{D}$ with $\mathfrak{P}\,\widetilde{\mathfrak{P}} = 1$ to the following unit dual quaternion:

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

Motion polynomials

Definition

A dual quaternion polynomial $\underline{\mathfrak{M}}(t) \in \mathbb{DH}[t]$ with a (nonzero) norm polynomial $\underline{\mathfrak{M}}(t)\,\widetilde{\underline{\mathfrak{M}}}(t) \in \mathbb{R}[t]$ is called motion polynomial.

Motion polynomials

Definition

A dual quaternion polynomial $\underline{\mathfrak{M}}(t) \in \mathbb{DH}[t]$ with a (nonzero) norm polynomial $\underline{\mathfrak{M}}(t)\widetilde{\underline{\mathfrak{M}}}(t) \in \mathbb{R}[t]$ is called motion polynomial.

We restrict ourselves to monic motion polynomials, motion passes through the idendity for $t = \infty$.

Motion polynomials

Definition

A dual quaternion polynomial $\underline{\mathfrak{M}}(t) \in \mathbb{DH}[t]$ with a (nonzero) norm polynomial $\underline{\mathfrak{M}}(t)\widetilde{\underline{\mathfrak{M}}}(t) \in \mathbb{R}[t]$ is called motion polynomial.

We restrict ourselves to monic motion polynomials, motion passes through the idendity for $t = \infty$.

Factorization

A generic monic motion polynomial $\underline{\mathfrak{M}}(t)$ of degree n admits at most $n!$ factorizations of the shape

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

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.

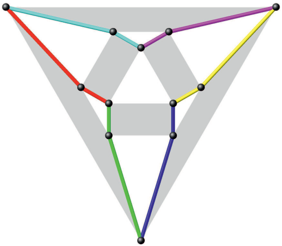
Factorization for our tool

Remark 1: Cubic motion polynomials

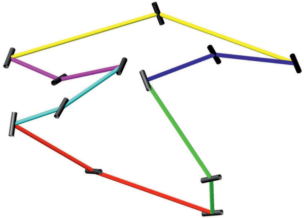
Remark 2: Six factorizations exist by Bennett flip, i.e.,

$$\begin{aligned}\underline{\mathfrak{M}}(t) &= (t-\underline{\mathfrak{A}}_1)(t-\underline{\mathfrak{A}}_2)(t-\underline{\mathfrak{A}}_3)\\ &= (t-\underline{\mathfrak{A}}_1)(t-\underline{\mathfrak{A}}'_2)(t-\underline{\mathfrak{A}}'_3)\\ &= (t-\underline{\mathfrak{A}}'_1)(t-\underline{\mathfrak{A}}''_2)(t-\underline{\mathfrak{A}}_3).\end{aligned}$$

Loops combination

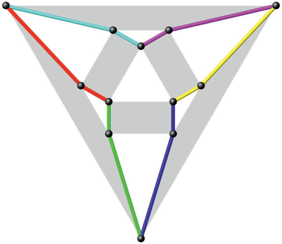


(a) Schematic

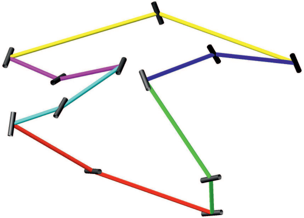


(b) Illustration

Loops combination

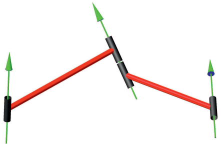


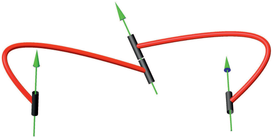
(a) Schematic



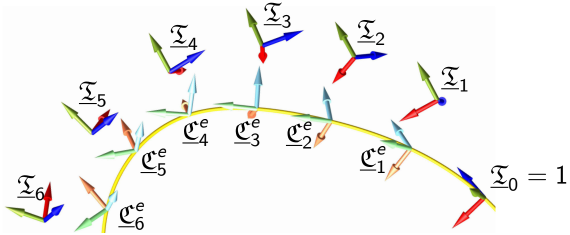
(b) Illustration

Links are not yet determined





Target poses and metric

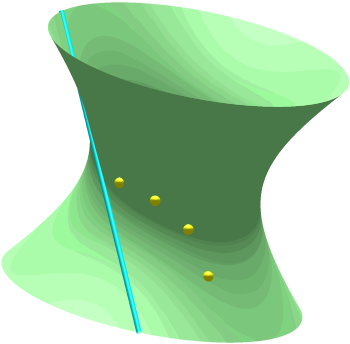


Assuming uniform mass distribution, the metric is simplified to:

$$\mathrm{dist}^2 := \frac{1}{6}\sum_{i=1}^{6} \|\sigma_1(v_i) - \sigma_2(v_i)\|^2,$$

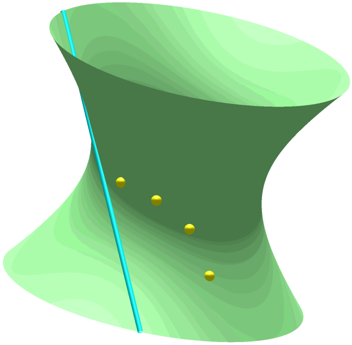
where v_i $(i = 1, \ldots, 6)$ are the six vertices of inertia ellipsoid, in our case, it is based on the moving object e.g. the shading element.

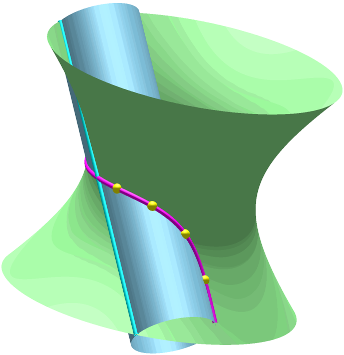
Four pose interpolation



(a) Input

Four pose interpolation

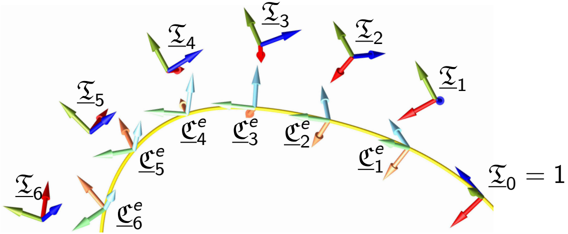

(a) Input


(b) Output

Four pose interpolation

- Two families solution (all 1-dof)
- Visit order might be wrong
- Exclude motions with singularities (based on a univariate polynomial)
- Rhino/Grasshopper tool Galapagos (applies evolutionary logic) for optimization

More than four poses: Evolution



Curve evolution algorithm

1. Initial guess:
   (a) *Four pose interpolation*
   (b) *Random quaternions*
2. Guiding poses:
   (a) *Closest pose projection:*
   (b) *Proportionally spaced:*
3. Curve Evolution:
   - *Initial stage:* Guiding poses by (2b) with back projection

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

   - *Middle stage:* Guiding poses by (2a) with back projection
   - *Final stage:* Guiding poses by (2a) without back projection

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

Statistic of curve evolution

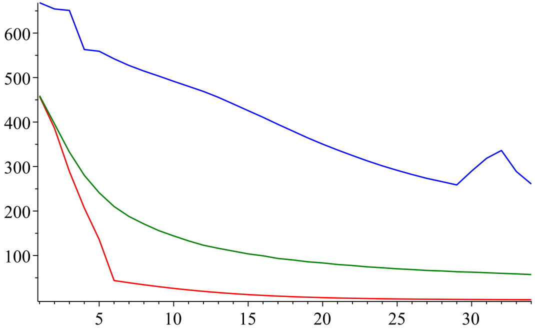
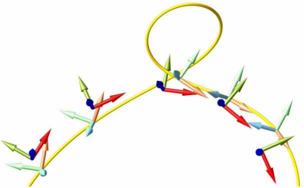


Figure: Statistic validation of the *curve evolution*.

Loop phenomenon of curve evolution



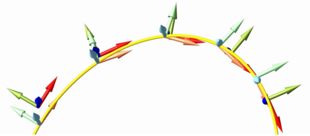
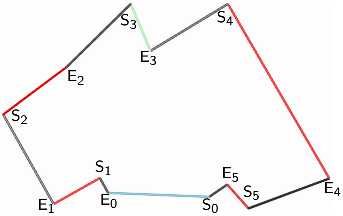


Figure: The loop phenomenon (left) can be avoided by a more costly modification of the curve evolution algorithm (right).

Curve evolution: comments

1. The four pose interpolation as a good initial guess.
2. Rhino/Grasshopper tool *Galapagos* for finding a good step size.
3. If the correct visit order is damaged or not achieved then we proceed with random initial guess.
4. Exclude rational cubic motions with singularities based on a univariate polynomial.

Realization ideas



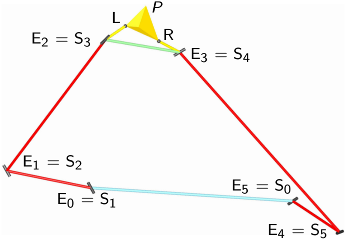


Figure: Line segments and end-effector

- Gray line segments denote rotation axes
- Colored line segments denote links

Offsetting

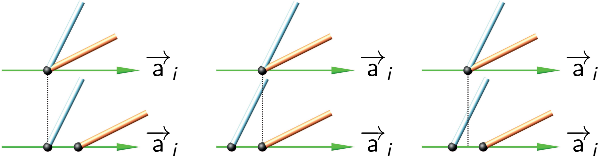


Figure: The three ways of offsetting.

Realization algorithm

1. Initialization :

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

2. Collision check: nine pairs of line-segments $(0,2)$, $(0,3)$, $(0,4)$, $(1,3)$, $(1,4)$, $(1,5)$, $(2,4)$, $(2,5)$, $(3,5)$
3. Search strategy: resulting $3^{6-f}-1$ linkages to be checked for collision
4. Link offsetting and thickening: 42 edge-edge collision checks

Realization

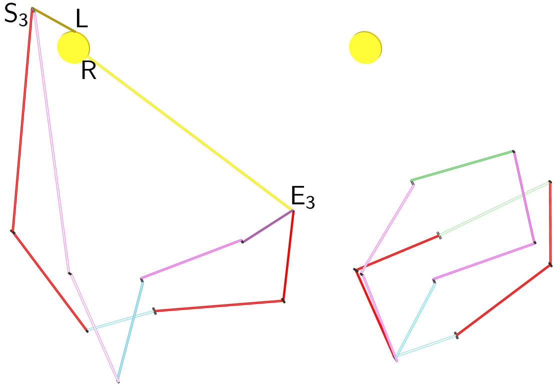


Figure: Realization integrated (yellow disc), or not.

Generating the link-design spaces

Post-processing algorithm:

(a) For each link, the user defines a potential link-design space (e.g. a cylinder).

(b) Each potential link-design space is trimmed by the other line-segments of the moving linkage.

(c) The boolean difference between each pair of the trimmed link-design spaces is performed over the complete motion cycle.

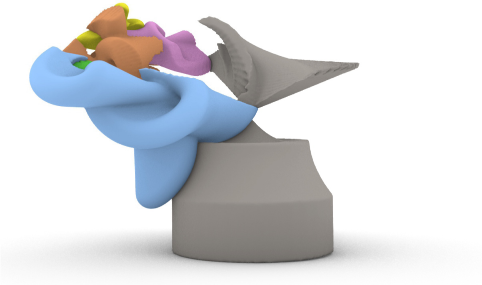
Generating the link-design spaces

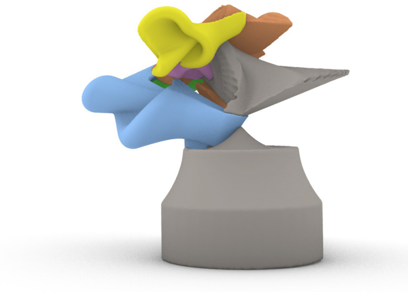
Post-processing algorithm:

(a) For each link, the user defines a potential link-design space (e.g. a cylinder).

(b) Each potential link-design space is trimmed by the other line-segments of the moving linkage.

(c) The boolean difference between each pair of the trimmed link-design spaces is performed over the complete motion cycle.





Loop grounding

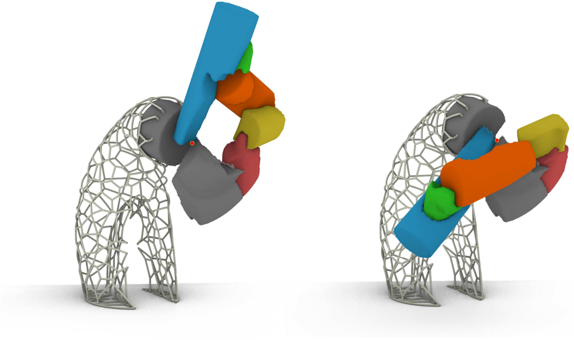


Figure: Extending one of the adjacent R-joints and fixing it to the ground.

Summary for linkage design

1. Realization Algorithm (succeeded in each of the 2000 validation tests)
2. Generation of collision-free link-design spaces (from line-segments)
3. Solution of the loop grounding problem

Design & Workflow

Difficulty

- Never been used
- No established design method
- How to proceed?

We did

- Linked master-level studio course and model making class in architecture (12 students)
- Task: use the presented tool to design a kinetic structure with focus on sculptural qualities and/or functional aspects (e.g. shading)
- Built complementary physical models

Chaotic Relay

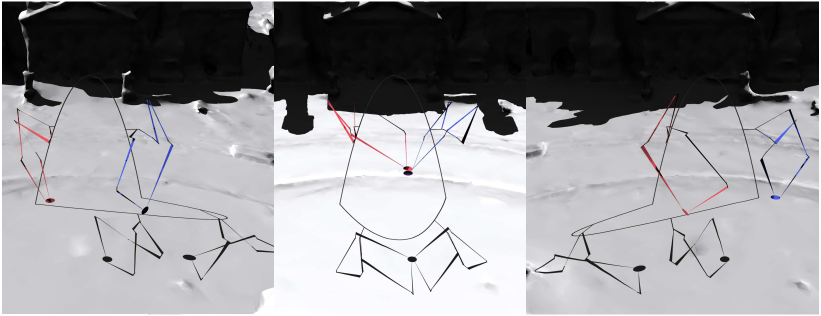


Figure: A series of target poses from the motion of the sun using evolution algorithm and symmetry.

Artificial Trees

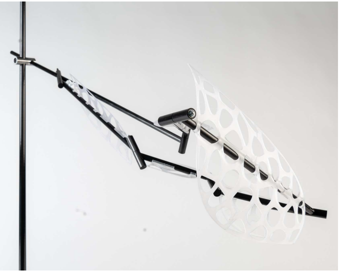


Figure: Visualization (left) and model photo (right) of an *Artificial Tree*

Conclusion

We gave an interactive tool (in Rhino/Grasshopper):

Conclusion

We gave an interactive tool (in Rhino/Grasshopper):

- Invertible paradoxic loops with six rotational joints
- Motion design via a 4 pose interpolation or motion evolution
- Self-collision-free over the complete motion cycle
- Used in a Master-level studio course
- Two results were fabricated

Conclusion

We gave an interactive tool (in Rhino/Grasshopper):

- Invertible paradoxic loops with six rotational joints
- Motion design via a 4 pose interpolation or motion evolution
- Self-collision-free over the complete motion cycle
- Used in a Master-level studio course
- Two results were fabricated

Future work

- extension for Bennett mechanisms and Goldberg linkages
- extension to paradoxic loop structures with prismatic (P) joints
- development for networks linkages
- classification of singularities of such 6R loops
- comparison of the *evolution algorithm* based different norms

Conclusion

We gave an interactive tool (in Rhino/Grasshopper):

- Invertible paradoxic loops with six rotational joints
- Motion design via a 4 pose interpolation or motion evolution
- Self-collision-free over the complete motion cycle
- Used in a Master-level studio course
- Two results were fabricated

Future work

- extension for Bennett mechanisms and Goldberg linkages
- extension to paradoxic loop structures with prismatic (P) joints
- development for networks linkages
- classification of singularities of such 6R loops
- comparison of the *evolution algorithm* based different norms

Thank you!