Motion Polynomials

Johannes Siegele

Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences (ÖAW)
Linz, Austria

April 5th 2024

Quaternions $\mathbb{H}$

Generated by 1 and the complex units $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ over $\mathbb{R}$:

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

Dual quaternions $\mathbb{DH}$

Generated by the additional dual unit ε:

$$p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k} + d_0\varepsilon + d_1\varepsilon\mathbf{i} + d_2\varepsilon\mathbf{j} + d_3\varepsilon\mathbf{k}$$

$$p + \varepsilon d, \qquad \text{where } p, d \in \mathbb{H}$$

Multiplication

Relations

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \quad \varepsilon^2 = 0,$$
$$\mathbf{i}\varepsilon = \varepsilon\mathbf{i}, \quad \mathbf{j}\varepsilon = \varepsilon\mathbf{j}, \quad \mathbf{k}\varepsilon = \varepsilon\mathbf{k}$$

	1	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
1	1	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
$\mathbf{i}$	$\mathbf{i}$	-1	$\mathbf{k}$	$-\mathbf{j}$
$\mathbf{j}$	$\mathbf{j}$	$-\mathbf{k}$	-1	$\mathbf{i}$
$\mathbf{k}$	$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	-1

Important Operators

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k},$$
$$h = p + \varepsilon d$$

- Conjugate quaternion: $q^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$
- Conjugate dual quaternion: $(p + \varepsilon d)^* = p^* + \varepsilon d^*$
- Quaternion norm: $qq^* = q_0^2 + q_1^2 + q_2^2 + q_3^2$
- Dual quaternion norm: $hh^* = pp^* + \varepsilon(pd^* + dp^*) \in \mathbb{D}$
- Vector part: $\vec{h} = \frac{1}{2}(h - h^*)$

Kinematic Image Space

The group of Euclidean displacements $\mathrm{SE}(3)$ is isomorphic to $\mathbb{DH}_0/\mathbb{R}^\times$, where

$$\mathbb{DH}_0 = \{p + \varepsilon d : pd^* + dp^* = 0, p \neq 0\}$$

Action of dual quaternions

Let $p + \varepsilon d \in \mathbb{DH}$ with $p \neq 0$ and $(x_1, x_2, x_3) \in \mathbb{R}^3$.

$$1 + \varepsilon x \mapsto \frac{1}{pp^*}(p - \varepsilon d) \cdot (1 + \varepsilon x) \cdot (p^* + \varepsilon d^*)$$

represents a Euclidean displacement, where

$$1 + \varepsilon x = 1 + \varepsilon(x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k}).$$

Quaternions and Spherical Rotations

Rotations

Any unit quaternion q represents a rotation around an axis through the origin in direction $\vec{q}$ via

$$q(1+\varepsilon x)q^*.$$

The angle φ of the rotation is given by

$$\cos\frac{\varphi}{2} = q_0, \quad \text{and} \quad \sin\frac{\varphi}{2} = |\vec{q}|$$

Continuous Rotations

A continuous rotation around an axis in direction $\vec{v}$ is given by

$$\cos(\varphi/2) + \sin(\varphi/2)\vec{v} = \sin(\varphi/2)\left(\frac{\cos(\varphi/2)}{\sin(\varphi/2)} + \vec{v}\right).$$

We can parametrize in two different ways:

- varying the angle $\varphi \in [0, 2\pi)$
- substitute $t = \cot(\varphi/2)$ and vary $t = (-\infty, \infty)$ in the expression $(t + \vec{v})$.

Dual Quaternion Polynomials

$$Q = q_n t^n + \ldots + q_1 t + q_0 = \sum_{i=0}^{n} q_i t^i,$$

with $q_0, \ldots, q_n \in \mathbb{DH}$.

- Conjugation: $Q^* = \sum_{i=1}^{n} q_i{}^* t^i$
- Norm polynomial: $QQ^* \in \mathbb{D}[t]$
- Right evaluation: $Q(h) := \sum_{i=1}^{n} q_i h^i$

Motion Polynomials

Dual quaternion polynomials Q with $QQ^* \in \mathbb{R}[t] \setminus \{0\}$

Rational Motions

Every motion polynomial represents a rational motion

Linear Polynomials

Linear motion polynomials $t - h$ for $h \in \mathbb{DH}$ parametrize

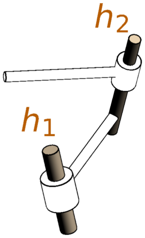
- Rotations
- Translations

Motion Factorization

For a motion polynomial $Q \in \mathbb{DH}[t]$, find linear polynomials $(t - h_1), \dots, (t - h_n)$ such that

$$Q = (t - h_1)(t - h_2) \dots (t - h_n)$$

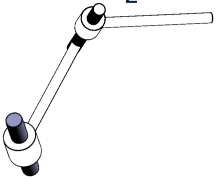
Construction of Mechanisms



$$Q = (t - h_1)(t - h_2)$$

Animation by Hans-Peter Schröcker

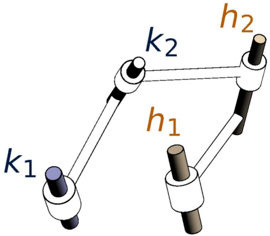
Construction of Mechanisms



$$Q = (t - k_1)(t - k_2)$$

Animation by Hans-Peter Schröcker

Construction of Mechanisms



$$Q = (t - h_1)(t - h_2) = (t - k_1)(t - k_2)$$

Animation by Hans-Peter Schröcker

Left Division

Let $C, P \in \mathbb{DH}[t]$ be two dual quaternion polynomials such that P is monic. Then there exist unique $Q, R \in \mathbb{DH}[t]$ with $\deg(R) < \deg(P)$ such that

$$C = QP + R$$

Algorithm

Input: Polynomials C, P
Output: Polynomials Q, R such that $C = QP + R$ and $\deg(R) < \deg(P)$

$n \leftarrow \deg(P)$
$R \leftarrow C$, $Q \leftarrow 0$
$m \leftarrow \deg(R)$
while $m \geq n$
  $r \leftarrow$ leading coefficient of R
  $Q \leftarrow Q + rt^{m-n}$
  $R \leftarrow R - rt^{m-n}P$
  $m \leftarrow \deg(R)$
end while
return Q, R

Theorem [Hegedüs, Schicho, Schröcker]

$(t - h)$ is a right factor of C if and only if $C(h) = 0$

Example

For $C = (t - \mathbf{i})(t - \mathbf{k})$ it holds $C = t^2 - (\mathbf{i} + \mathbf{k})t - \mathbf{j}$, hence

$$C(\mathbf{k}) = \mathbf{k}^2 - (\mathbf{i} + \mathbf{k})\mathbf{k} - \mathbf{j} = -1 + \mathbf{j} + 1 - \mathbf{j} = 0$$

$$C(\mathbf{i}) = \mathbf{i}^2 - (\mathbf{i} + \mathbf{k})\mathbf{i} - \mathbf{j} = -1 - \mathbf{j} + 1 - \mathbf{j} = -2\mathbf{j} \neq 0$$

Theorem [Hegedüs, Schicho, Schröcker]

Let $C = P + \varepsilon D$ be a motion polynomial and M a monic quadratic factor of $CC^* \in \mathbb{R}[t]$ such that M does not divide P. Then there exists a unique $h \in \mathbb{DH}$ such that $(t - h)$ is a right factor of C and $M = (t - h^*)(t - h)$.

How to find a right factor

Given: A motion polynomial C

Compute: The real polynomial CC^*

Compute: A monic quadratic factor $M \in \mathbb{R}[t]$ of CC^*

Division with remainder: $C = QM + R$, with $R = r_1 t + r_0$

Set: $h = -\frac{{r_1}^* r_0}{r_1 {r_1}^*}$. Then $(t - h)$ is a right factor of C

Sketch of proof

Given is a monic quadratic factor M of CC^*. We can compute $C = QM + R$. It holds

$$
\begin{aligned}
CC^* &= (QM + R)(QM + R)^* \\
&= M(QQ^*M + QR^* + RQ^*) + RR^*
\end{aligned}
$$

Thus, M is a factor of RR^*
The assumptions of the theorem ensure, that the leading coefficient r_1 of R is invertible, thus we can compute the unique zero h of R.
From $R^*R = {r_1}^* r_1 (t-h)^*(t-h)$ it follows $M = (t-h)^*(t-h)$ and we get

$$
C = Q(t-h)^*(t-h) + r_1(t-h) = \left(Q(t-h)^* + r_1\right)(t-h)
$$

Some Remarks

- We focus on “reduced” motion polynomials C
- This yields an algorithm for the factorization of “generic” motion polynomials $C = P + \varepsilon D$
- The obtained factorization depends on the chosen quadratic factor M in each step
- In general, there exist $n!$ different factorizations
- This algorithm relies on the fact, that we can find a factorization of the real norm polynomial. This, however, is by far no easy task
- If the motion polynomial is non-generic, there might exist no or even infinitely many factorizations

Example

The polynomial $C = t^2 + 1 + \varepsilon(a\mathbf{i} + b\mathbf{j}t)$ corresponds to a translation along an ellipse. It does not admit a factorization unless $a = b$, in which case there are infinitely many factorizations.

The non-generic case

Let $C = cP + \varepsilon D$, where $c \in \mathbb{R}[t]$

- The algorithm still works, if c only has real roots $a \in \mathbb{R}$ of multiplicity one. We can take $M = (t - a)^2$
- If c has real roots of higher multiplicity, there does not exist a factorization:

$$(\lambda - \varepsilon d_1)(\lambda - \varepsilon d_2) = \lambda(\ldots)$$

- If c only has complex roots, there exists a criterion for factorizability
- In this case, there exists a real polynomial $S \in \mathbb{R}[t]$ with $\deg S \leq \deg c$ such that SC admits a factorization.

Plücker lines

- A line in direction $v \in \mathbb{R}^3$ through the point $p \in \mathbb{R}^3$ can be represented by its Plücker coordinates $(v, p \times v) \in \mathbb{P}^5$
- A half-turn around an axis with Plücker coordinates (v, m) is given by $(v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}) - \varepsilon(m_1\mathbf{i} + m_2\mathbf{j} + m_3\mathbf{k}) = v - \varepsilon m$
- The axis of the rotation given by $t - h$ has Plücker coordinates $\vec{h}_\varepsilon$

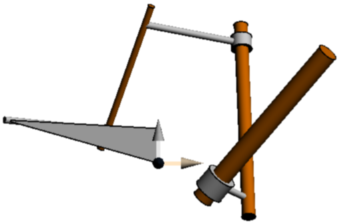
Constructing a Bennett Mechanism

Let C be a quadratic motion polynomial with two factorizations $C = (t - h_1)(t - h_2) = (t - k_1)(t - k_2)$.

- We have two fixed axes given by $\vec{h_1}$, $\vec{k_1}$
- The two moving axes can be obtained by rotating the lines given by $\vec{h_2}$ and $\vec{k_2}$ around $\vec{h_1}$ and $\vec{k_1}$, respectively, i.e.

$$(t - h_1)\vec{h_2}(t - {h_1}^*)$$
$$(t - k_1)\vec{k_2}(t - {k_1}^*)$$

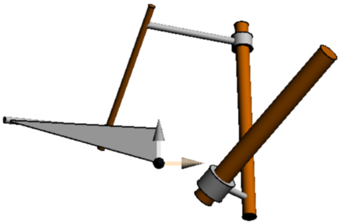
An overconstrained 6R



$$C = (t - h_1)(t - h_2)(t - h_3)$$
$$= (t - k_1)(t - k_2)(t - k_3)$$

Animation by Hans-Peter Schröcker

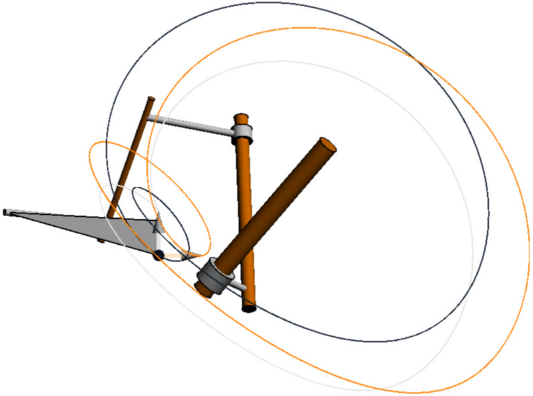
An overconstrained 6R



$$C = (t - h_1)(t - h_2)(t - h_3)$$
$$= (t - k_1)(t - k_2)(t - k_3)$$

Animation by Hans-Peter Schröcker

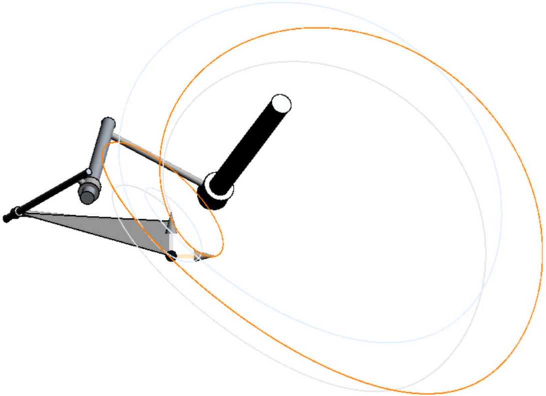
An overconstrained 6R



$$C = (t - h_1)(t - h_2)(t - h_3)$$
$$= (t - k_1)(t - k_2)(t - k_3)$$

Animation by Hans-Peter Schröcker

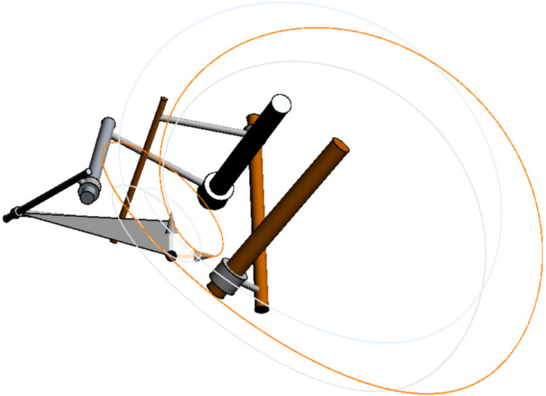
An overconstrained 6R



$$C = (t - h_1)(t - h_2)(t - h_3)$$
$$= (t - k_1)(t - k_2)(t - k_3)$$

Animation by Hans-Peter Schröcker

An overconstrained 6R



$$C = (t - h_1)(t - h_2)(t - h_3)$$
$$= (t - k_1)(t - k_2)(t - k_3)$$

Animation by Hans-Peter Schröcker

3D-printed Linkages

Python Package

By Daniel Huczala:
daniel.huczala@uibk.ac.at

Package Documentation:
rational-linkages.readthedocs.io