

Computer-Aided Design of Rational Motions for 4R and 6R Spatial Mechanism Synthesis

Daniel Huczala, Severinas Zube, Martin Pfurner, Johannes Siegele, and Frank C. Park

Abstract This paper focuses on geometric methods for generating rational motions used in the design of single-loop rational linkages, 1-degree-of-freedom mechanisms that can execute prescribed spatial tasks. Building on established rational motion synthesis methods, we introduce a new interpolation scheme for seven 3D points based on cubic quaternionic Bézier curves. The resulting motion directly allows synthesis of generating mechanisms via known factorization technique. To support engineering practice, we provide open-source CAD tools that implement also the other methods and provide fast visual evaluation of motion generation and mechanism synthesis.

1 Introduction

In robot kinematics, the single-loop one degree of freedom (1-DoF) mechanism synthesis problem for a prescribed set of points or poses, known as path generation or motion generation, has attracted interest for decades. The 1-

D. Huczala
Robotics Laboratory, Seoul National University, Seoul, Korea
e-mail: daniel.huczala@snu.ac.kr

S. Zube
Institute of Computer Science, Department of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania

M. Pfurner
Unit of Geometry and Surveying, University of Innsbruck, Innsbruck, Austria

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

F. C. Park
Robotics Laboratory, Seoul National University, Seoul, Korea

DoF property provides straightforward and energy-efficient design [?], but its *spatial* applications are more limited than those in the planar or spherical cases. One of the reasons is that in spatial kinematics the generation of 1-DoF motion is more challenging. There is no meaningful distance metric in three-dimensional special Euclidean group $SE(3)$ [?], which, for example, hinders numerical optimization techniques. Therefore, researchers have been investigating algebraic ways for motion generation, and this study also utilizes these findings.

In case of planar four-bar mechanisms, the motion generation is also known as the Burmester Problem and was solved in multiple ways, for example with five two-dimensional input poses [?]. The planar problem is closely related to the spherical case, and the results presented share similarities [?]. Although spatial interpolation schemes have been extensively studied [?], not all are applicable here, as our interest is in low-degree continuous motions that guide the origin – or the entire moving frame – through a given set of points or poses, enabling the synthesis of a 1-DoF mechanism *via* factorization theory. As is known, there exists only one spatial closed-chain four-bar linkage class (the Bennett mechanism), and there are algebraic methods that construct the Bennett motion that can achieve three given poses [?] or five points [?]. The resulting motion is parameterized by a rational function and the parameter corresponds directly to the driving joint angle [?]. The design parameters of the mechanism can then be obtained using the theory of *rational motion factorization* [?], which decomposes the motion curve into linear factors that represent rotations (materialized by revolute joints). This powerful algebraic algorithm can also be applied to motions of higher degree, as applied in [?] for four arbitrary poses in space interpolated by a cubic motion, which corresponds to 1-DoF six-bar mechanisms.

In this paper, we extend the motion generation methodology of [?], provide updated explicit formulas for the case of interpolation through five points, and introduce the interpolation scheme for seven points in a similar manner. The output motions are returned in a factorizable form, i.e. the four- and six-bar mechanism synthesis can be performed immediately. Additionally, an open-source implementation is provided not only for these new interpolation methods, but also for the algorithms of [?][?], including mechanism synthesis using the factorization method, all as an interactive CAD tool (see video in [?]).

The paper is organized as follows: Sect. 2 covers mathematical preliminaries; Sect. 3 reviews interpolation methods for three and four poses before addressing the five- and seven-point cases; and Sect. 3.6 provides the implementation details and references to the supplementary material.

2 Mathematical Preliminaries

To represent proper rigid body motions of the special Euclidean group $SE(3)$, we use dual quaternions $\mathbb{DH} = \{p + \varepsilon q \mid p, q \in \mathbb{H}, \varepsilon^2 = 0\}$, alternatively known as Study parameters. They extend regular quaternions $\mathbb{H} \cong \mathbb{R}^4$ in the following way. Given quaternions p, q , a dual quaternion c can be written as

$$c = p + \varepsilon q = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k} + \varepsilon(q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}), \quad (1)$$

where ε is the dual unit that squares to zero. The conjugation of dual quaternions is defined as $c^* := p^* + \varepsilon q^*$ and the ε -conjugation is defined as $c_\varepsilon := p - \varepsilon q$.

The elements of $\mathbb{DH}$ which lie in the *Study quadric* $S \subset \mathbb{DH}$

$$S = \{p + \varepsilon q \mid pq^* + qp^* = 2(p_0q_0 + p_1q_1 + p_2q_2 + p_3q_3) = 0\} \quad (2)$$

represent a rigid body displacement. For more details, including the exceptional space on S or dual quaternion multiplication rules, see [?, Chap. 13].

We will identify 3D points $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ with imaginary quaternions $\mathbf{x} = x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \in \mathbf{Im}(\mathbb{H})^1$, i.e. quaternions with zero first entry $x_0 = 0$. Using this identification, we associate a point $p + \varepsilon q$ on the Study quadric with a 3D point using the map π

$$\pi(p + \varepsilon q) = qp^*/pp^* = qp^{-1}, \quad (3)$$

where $p^{-1} = p^*/pp^*$ is an inverse quaternion. The action of a dual quaternion $c = p + \varepsilon q$ on a point x is then

$$x \mapsto p\mathbf{x}p^{-1} + qp^{-1}. \quad (4)$$

Note that the first summand corresponds to the rotational part of the displacement, while the second summand describes the translational part. Therefore, (3) maps a dual quaternion to the image of the origin under this displacement, i.e., the map π extracts the coordinates of the origin of the displaced frame.

A one-parametric motion corresponds to a curve of a form $C(t) \in S$ and can be written in the form of a dual quaternion polynomial $C = p + \varepsilon q \in \mathbb{DH}[t]$. It describes a rigid body motion if and only if $CC^* \in \mathbb{R}[t] \setminus \{0\}$, i.e. it lies on the Study quadric ($C(t) \in S$) or equivalently, $qp^*, qp^{-1} \in \mathbf{Im}(\mathbb{H}[t])$. The trajectories of points in these motions follow rational trajectories parameterized by t . As mentioned above, $C(t)$ can be factorized [?] to synthesize mechanisms.

¹ For some conventions, e.g. [?], $\mathbf{x} = -\frac{1}{2}(x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k})$ is required.

3 Methods to Construct Rational Motions

3.1 Quadratic Motion through 3 Poses

Three poses $c_{0..2}$ define a plane in the space of dual quaternions uniquely. This plane intersects S in (2) in a *conic*, which is the quadratic motion curve to be constructed. Such curves were classified and studied extensively in [?]. According to [?] the conics passing through $c_{0..2}$ can be parameterized in the form

$$C(t) = \alpha c_0 + (c_1 - \alpha c_0 - \beta c_2)t + \beta c_2 t^2, \quad (5)$$

where α and β must be chosen so that the entire curve lies on S . Substitution of $C(t)$ into S produces a quartic equation $M(t)$ in t . Due to the parameterization in (5) the coefficients of t^0 and t^4 in $M(t)$ vanish and it factorizes into $M(t) = t(t-1)G(t)$ with a linear polynomial $G(t)$. This must vanish independently of t and, therefore, its coefficients yield a linear system of equations for α and β .

3.2 Cubic Motion through 4 Poses

For interpolating four poses with a cubic motion, we give a short summary of the algorithm presented in [?]. Four poses span a three-space in the dual quaternions which intersects S in a quadric containing all four poses $c_{0..3}$. An interpolating cubic motion exists if this quadric contains straight lines. Assuming such lines exist, the two straight lines through $c_0 = 1$ must contain two vectorial² dual quaternions k_1 and k_2 . Any twisted cubic on the given quadric must intersect one of these two lines in two points. Thus, let us assume that it intersects the line $t - k_1$ twice and define $c_4 = \lambda - k_1$ for some $\lambda \in \mathbb{R}$. The parameter values for which the cubic interpolates $c_0, \dots, c_4$ are given by the parameters at which the rulings through the given poses intersect the line parametrized by $t - k_1$. From this condition, we can compute $t_0 = \infty, t_1, \dots, t_4 = \lambda$. With these parameter values, we can compute the Lagrange polynomial C that interpolates $\lambda_i c_i$ at t_i and indeterminates λ_i for $i = 0, \dots, 3$. To also interpolate the last pose c_4 , we can solve $C(t_4) = \lambda_4 c_4$ for λ_i . Note that the algorithm could also use k_2 as input which gives a *second solution*. Further, the obtained polynomial depends on the chosen λ , i.e., the algorithm of [?] results in two sets of 1-parameter families of cubic motions. Control of these degrees of freedom is also implemented in the CAD tool presented later.

² These are dual quaternions with both p and $q \in \mathbf{Im}(\mathbb{H})$.

3.3 Motion through $2n + 1$ Points

In contrast to interpolation through poses, the next sections interpolate via 3D points. The construction via quaternionic Bézier curves is applied, i.e., with control points in $\mathbb{DH}$ lying on S . The quaternionic weights are specified in such a way that the resulting rational curve lies on Study quadric S and is a proper continuous motion.

We choose arbitrary $2n + 1$ 3D points a_i , $i = 0, 1, \dots, 2n$, which are the points *to be interpolated*. Let $n \geq 1$ be an integer, $T = [t_0, \dots, t_n]$ be $n + 1$ distinct real values. Using

$$f_i(t) = (t - t_0)(t - t_1) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_n), \quad i = 0, 1, \dots, n, \quad (6)$$

we define quaternionic polynomials of degree n and a motion polynomial $C(t)$

$$p(t) = \sum_{i=0}^n w_{2i} f_i(t), \quad q(t) = \sum_{i=0}^n a_{2i} w_{2i} f_i(t), \quad C(t) = p(t) + \varepsilon q(t), \quad (7)$$

where weights $w_0, w_2, \dots, w_{2n}$ are quaternions which will be determined later. First of all, note that

$$\pi(C(t_i)) = a_{2i} \in \mathbf{Im}(\mathbb{H}) = \mathbb{R}^3, \quad i = 0, 1, \dots, n, \quad (8)$$

for arbitrary weights w_{2i} , $i = 0, 1, \dots, n$ (because $f_j(t_i) = 0$, $j \neq i$). This shows that $C(t)$ already interpolates the points a_{2i} for $i = 0, \dots, n$. To ensure that also the remaining points are interpolated, we set up a system of equations to determine the appropriate weights w_i . For this, let $T_s = [s_1, \dots, s_n]$ be different real parameters such that $T_s \cap T = \emptyset$ and for which the points a_{2i+1} are interpolated.

We consider the linear system $\pi(C(s_j)) = a_{2j-1}$, $j = 1, \dots, n$. Explicitly this linear system is equivalent to

$$\sum_{i=1}^n (a_{2i} - a_{2j-1}) w_{2i} f_i(s_j) = (a_{2j-1} - a_0) w_0 f_0(s_j), \quad j = 1, \dots, n. \quad (9)$$

Without loss of generality, we may assume that $w_0 = 1$. The quaternion linear system (9) with n equations and n unknowns $w_2, w_4, \dots, w_{2n}$ has a unique solution if the corresponding matrix A is not singular. Theorem 1 in [?] claims that for a *general* choice of points a_i , $i = 0, \dots, 2n + 1$ there is a unique solution.

It is left to show that the motion obtained $C(t)$ meets the Study condition. For this, we need to show $pq^* + qp^* = 0$. This polynomial is of degree $2n$ and is the primal part of $\pi(C(t))$. Hence, it is fulfilled for the $2n + 1$ parameter values $t_0, \dots, t_n$ and $s_1, \dots, s_n$. Therefore, we have a polynomial of degree $2n$ which vanishes for $2n + 1$ parameters, thus has to be 0.

In the following sections, we present explicit formulas for cases $n = 2, 3$, which result in quadratic and cubic motions.

3.4 Quadratic Motion through 5 Points

For the quadratic motion, we fix five 3D points (via points) $a_0, \dots, a_4$ and parameter values³ (via times) $T = [0, \frac{1}{2}, 1]$ and $T_s = [\frac{1}{4}, \frac{3}{4}]$. The linear system defined by (9) can be solved (see [?]). Weights w_2, w_4 are

$$w_2 = (-9 d_{41}^{-1} d_{21} - 3 d_{43}^{-1} d_{23})^{-1} (9 d_{41}^{-1} d_{10} - d_{43}^{-1} d_{30}) w_0, \quad (10)$$

$$w_4 = (-d_{21}^{-1} d_{41} - 3 d_{23}^{-1} d_{43})^{-1} (3 d_{21}^{-1} d_{10} + d_{23}^{-1} d_{30}) w_0, \quad (11)$$

where $d_{ij} = a_i - a_j$. Choosing $w_0 = 1$ we compute the motion $C(t)$ using (7).

To obtain the curve $C(t)$ in Bézier form, we express the interpolation basis $f_{0..2}$ by quadratic Bernstein polynomials $\beta_i(t) = \binom{2}{i}(1-t)^{2-i}t^i$, $i = 0, 1, 2$

$$f_0 = \frac{1}{2}\beta_0 - \frac{1}{4}\beta_1, \quad f_1 = -\frac{1}{2}\beta_1, \quad f_2 = -\frac{1}{4}\beta_1 + \frac{1}{2}\beta_2. \quad (12)$$

We substitute (12) into the motion $C = p + \varepsilon q$

$$\begin{aligned} p &= w_0 f_0 + w_2 f_1 + w_4 f_2 = \\ &= \frac{1}{2}w_0\beta_0 + \left(-\frac{1}{4}w_0 - \frac{1}{2}w_2 - \frac{1}{4}w_4\right)\beta_1 + \frac{1}{2}w_4\beta_2 = u_0\beta_0 + u_1\beta_1 + u_2\beta_2, \\ q &= a_0 w_0 f_0 + a_2 w_2 f_1 + a_4 w_4 f_2 = a_0 \frac{1}{2}w_0\beta_0 - \\ &\quad - \frac{1}{4}(a_0 w_0 + 2a_2 w_2 + a_4 w_4)\beta_1 + \frac{1}{2}a_4 w_4\beta_2 = p_0 u_0\beta_0 + p_1 u_1\beta_1 + p_2 u_2\beta_2. \end{aligned}$$

So we obtain the following weights and points

$$\begin{aligned} u_0 &= \frac{1}{2}w_0, \quad u_1 = -\frac{1}{4}(w_0 + 2w_2 + w_4), & u_2 &= \frac{1}{2}w_4, \\ p_0 &= a_0, \quad p_1 = -\frac{1}{4}(a_0 w_0 + 2a_2 w_2 + a_4 w_4)(u_1)^{-1}, & p_2 &= a_4. \end{aligned}$$

Then the motion $C(t)$ can be presented in the Bézier from

$$C(t) = \left(\sum_{i=0}^2 u_i \beta_i \right) + \varepsilon \left(\sum_{i=0}^2 p_i u_i \beta_i \right) \quad (13)$$

³ The order matters as a_0 will be interpolated at $t = 0$, a_1 at $t = \frac{1}{4}$, a_2 at $t = \frac{1}{2}$, etc.

with the 8-tuples $u_i + \varepsilon p_i u_i \in \mathbb{DH}, i = 0, 1, 2$ being the three homogeneous control points of the 8-dimensional quadratic Bézier curve that lies on S and can be factorized to synthesize a mechanism.

3.5 Cubic Motion through 7 Points

For the cubic motion we fix seven 3D points $a_0, \dots, a_6$ and parameter values $T = [0, \frac{1}{3}, \frac{2}{3}, 1]$ and $T_s = [\frac{1}{6}, \frac{1}{2}, \frac{5}{6}]$, and we will need to determine four control points and their weights. An example together with the resulting motion is shown in Fig. 1 on the left. The linear system defined by (9) can be written as

$$l_j : c_{j,2} w_2 + c_{j,4} w_4 + c_{j,6} w_6 = c_{j,8} w_0, \quad (j = 1, 2, 3), \quad (14)$$

where the coefficients $c_{j,i}$ are given by

$$\begin{pmatrix} c_{1,2} & c_{1,4} & c_{1,6} & c_{1,8} \\ c_{2,2} & c_{2,4} & c_{2,6} & c_{2,8} \\ c_{3,2} & c_{3,4} & c_{3,6} & c_{3,8} \end{pmatrix} = \begin{pmatrix} 15 d_{21} & 5 d_{41} & 3 d_{61} & -15 d_{10} \\ 9 d_{23} & 9 d_{43} & -3 d_{63} & 3 d_{30} \\ -5 d_{25} & -15 d_{45} & 15 d_{65} & -3 d_{50} \end{pmatrix}. \quad (15)$$

with $d_{ij} = a_i - a_j$. We assume that $w_0 = 1$ and eliminate the unknown w_2 using $c_{1,2}^{-1} l_1 - c_{2,2}^{-1} l_2$ and $c_{1,2}^{-1} l_1 - c_{3,2}^{-1} l_3$, which can be written as

$$m_j : e_{j,4} w_4 + e_{j,6} w_6 = e_{j,8}, \quad (j = 2, 3) \quad (16)$$

$$\begin{pmatrix} e_{2,4} & e_{2,6} & e_{2,8} \\ e_{3,4} & e_{3,6} & e_{3,8} \end{pmatrix} = \begin{pmatrix} h_{4,2,4} & h_{6,2,6} & h_{8,2,8} \\ h_{4,3,4} & h_{6,3,6} & h_{8,3,8} \end{pmatrix} \quad (17)$$

where $h_{i,j,k} = E(c_{1,2}, c_{1,i}, c_{j,2}, c_{j,k})$, with $E(a, b, c, d) = a^{-1}b - c^{-1}d$ for arbitrary quaternions a, b, c, d . Similarly, we eliminate w_6 and w_4 from equations m_2 and m_3 by taking $e_{2,6}^{-1} m_2 - e_{3,6}^{-1} m_3$ leading to $r_{4,4} w_4 = r_{4,8}$ and $e_{2,4}^{-1} m_2 - e_{3,4}^{-1} m_3$ leading to $r_{6,6} w_6 = r_{6,8}$, where

$$r_{4,4} = E(e_{2,6}, e_{2,4}, e_{3,6}, e_{3,4}), \quad r_{4,8} = E(e_{2,6}, e_{2,8}, e_{3,6}, e_{3,8}), \quad (18)$$

$$r_{6,6} = E(e_{2,4}, e_{2,6}, e_{3,4}, e_{3,6}), \quad r_{6,8} = E(e_{2,4}, e_{2,8}, e_{3,4}, e_{3,8}), \quad (19)$$

$$w_4 = r_{4,4}^{-1} r_{4,8}, \quad w_6 = r_{6,6}^{-1} r_{6,8}. \quad (20)$$

Using (14) we compute w_2 by $w_2 = c_{1,8} - c_{1,4} w_4 - c_{1,6} w_6$. Therefore, we found a parametrization of a cubic curve by the formula (7). The parametrization $\pi(C(t))$ interpolates points $a_0, \dots, a_6$ at the parameter values $\frac{i}{6}, i = 0, \dots, 6$.

To present the curve in Bézier form, we need to convert the interpolation basis $[f_0, \dots, f_3]$ to Bézier basis $[\beta_0, \dots, \beta_3]$, where $\beta_i(t) = \binom{3}{i} (1-t)^{3-i} t^i$ as:

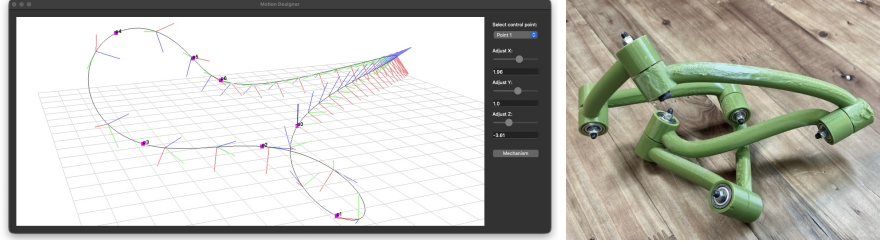


Fig. 1: Left: Motion Designer – an interactive plot of a 1-DoF motion that interpolates 7 given points (purple); right: example of a produced 6R prototype.

$$f_0 = -\frac{2}{9}\beta_0 + \frac{5}{27}\beta_1 - \frac{2}{27}\beta_2, \quad f_1 = \frac{2}{9}\beta_1 - \frac{1}{9}\beta_2, \quad (21)$$

$$f_2 = \frac{1}{9}\beta_1 - \frac{2}{9}\beta_2, \quad f_3 = \frac{2}{27}\beta_1 - \frac{5}{27}\beta_2 + \frac{2}{9}\beta_3. \quad (22)$$

Now we may write the interpolated cubic (defined by the formulas (7)) in Bézier form in the same way (after substitution in p, q) as in the previous section

$$C(t) = (u_0\beta_0 + \dots + u_3\beta_3) + \varepsilon(p_0u_0\beta_0 + \dots + p_3u_3\beta_3), \quad (23)$$

where

$$\begin{aligned} u_0 &= -\frac{2}{9}w_0, \quad u_1 = \frac{5}{27}w_0 + \frac{2}{9}w_2 + \frac{1}{9}w_4 + \frac{2}{27}w_6, \\ u_2 &= -\frac{2}{27}w_0 - \frac{1}{9}w_2 - \frac{2}{9}w_4 - \frac{5}{27}w_6, \quad u_3 = \frac{2}{9}w_6, \\ p_0 &= a_0, p_1 = \left(\frac{5}{27}a_0w_0 + \frac{2}{9}a_2w_2 + \frac{1}{9}a_4w_4 + \frac{2}{27}a_6w_6 \right) u_1^{-1}, \\ p_2 &= \left(-\frac{2}{27}a_0w_0 - \frac{1}{9}a_2w_2 - \frac{2}{9}a_4w_4 - \frac{5}{27}a_6w_6 \right) u_2^{-1}, p_3 = a_6. \end{aligned}$$

3.6 Implementation

For the kinematics community, we implemented all methodologies as an interactive CAD module called Motion Designer in the Rational Linkages library [?] version 2.5, an open-source Python package for synthesis and rapid prototyping of these mechanisms. Some crucial calculations were compiled using the Rust language to provide fast rendering with Qt6 software. The visual CAD interface is shown in Fig. 1 with a 6R prototype designed using this

methodology. For more details and source code investigation, see the supplementary material [?].

4 Conclusion

This study introduced a way to interactively construct rational motions interpolating three or four poses and five or seven points. In addition to state-of-the-art, closed-form solutions for the latter interpolations of points with rational curves in $SE(3)$ were presented, along with the implementation providing visual CAD software. The output motions serve as the key concept for the synthesis of 1-DoF single-loop mechanisms using rational motion factorization [?], resulting in spatial four- and six-bar mechanisms. The algorithms are part of the Rational Linkages library [?], version 2.5 or higher. The interactive use case is shown in the accompanying video together with web-links to the source code, documentation, and tutorials, and can be found in [?].

Acknowledgements This work was supported in part by IITP-MSIT grant RS-2021-II212068 (SNU AI Innovation Hub), IITP-MSIT grant 2022-220480, RS-2022-II220480 (Training and Inference Methods for Goal Oriented AI Agents), MSIT (Ministry of Science, ICT), Korea, under the Global Research Support Program in the Digital Field program (RS-2024-00436680) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), KIAT grant P0020536 (HRD Program for Industrial Innovation), SRRC NRF grant RS-2023-00208052, SNU-AIIS, SNU-IPAI, SNU-IAMD, SNU BK21+ Program in Mechanical Engineering, SNU Institute for Engineering Research, and Microsoft Research Asia. We gratefully acknowledge the support of the bilateral Austrian–Korean project “Surface generation via integrable evolution of curves,” which is funded by OeAD-GmbH (Austria’s Agency for Education and Internationalization, project no. KR 02/2025), and the National Research Foundation of Korea (project no. RS-2025-1435299). This work was also supported by the Vizerektorat für Forschung of the University of Innsbruck.

References

1. Bottema, O., Roth, B.: Theoretical Kinematics. North-Holland Publ. Comp. (1979)
2. Brunnthaler, K., Schröcker, H.P., Husty, M.: A new method for the synthesis of Bennett mechanisms. Int. Workshop on Computational Kinematics (2005)
3. Ge, Q.J.: Motion Interpolation, p. 1–7. Springer Berlin Heidelberg (2021). doi:10.1007/978-3-642-41610-1_123-1
4. Hamann, M.: Line-symmetric motions with respect to reguli. Mech. Mach. Theory **46**(7), 960–974 (2011). doi:10.1016/j.mechmachtheory.2011.02.004
5. Hayes, M.J.D., Zsombor-Murray, P.J.: Solving the burmester problem using kinematic mapping. In: 27th Biennial Mechanisms and Robotics Conference (2002)
6. Hegedüs, G., Schicho, J., Schröcker, H.P.: Factorization of rational curves in the Study quadric. Mech. Mach. Theory **69**, 142–152 (2013)
7. Hegedüs, G., Schicho, J., Schröcker, H.P.: Four-pose synthesis of angle-symmetric 6r linkages. J. Mech. Robot. **7**(4) (2015). doi:10.1115/1.4029186

8. Huczala, D., Mair, A., Postulka, T.: Direct kinematics, inverse kinematics, and motion planning of 1-dof rational linkages. *Mech. Mach. Theory* **213** (2025)
9. Huczala, D., Siegele, J., Thimm, D.A., Pfurner, M., Schröcker, H.P.: Rational linkages: From poses to 3d-printed prototypes. In: *Advances in Robot Kinematics 2024*. ARK 2024. Springer (2024). doi:10.1007/978-3-031-64057-5_27
10. Huczala, D., Zube, S., Pfurner, M., Siegele, J., Park, F.C.: CAD of rational motions (suppl. material) (2026). URL <https://doi.org/10.5281/zenodo.18081828>
11. Mullineux, G.: Atlas of spherical four-bar mechanisms. *Mech. Mach. Theory* **46**(11), 1811–1823 (2011). doi:10.1016/j.mechmachtheory.2011.06.001
12. Park, F.C.: Distance metrics on the rigid-body motions with applications to mechanism design. *J. Mech. Des.* **117**(1), 48–54 (1995). doi:10.1115/1.2826116
13. Poštulka, T., Huczala, D., Orzechowski, G., Mikkola, A., Bobovský, Z., Huňady, R.: Reducing energy consumption in robotic processes using a single-dof bennett mechanism. *IEEE Access* (2025). doi:10.1109/access.2025.3646349
14. Zube, S.: Interpolation method for quaternionic-bezier curves. *Lietuvos matematikos rinkinys* **59**, 13–18 (2018). doi:10.15388/lmr.a.2018.03