Four-Pose Synthesis of Angle-Symmetric 6R Linkages

Gábor Hegedüs
Applied Mathematical Institute, Antal Bejczy Center for Intelligent Robotics, Obuda University
hegedus.gabor@nik.uni-obuda.hu

Josef Schicho
Johan Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences
josef.schicho@oeaw.ac.at

Hans-Peter Schröcker
Unit Geometry and CAD, University Innsbruck,
hans-peter.schroecker@uibk.ac.at

ABSTRACT

We use the recently introduced factorization theory of motion polynomials over the dual quaternions for the synthesis of closed kinematic loops with six revolute joints that visit four prescribed poses. Our approach admits either no or a one-parametric family of solutions. We suggest strategies for picking good solutions from this family.

1 Introduction

In [1], Brunnthaler et al. presented a method for synthesizing closed kinematic chains of four revolute joints whose coupler can visit three prescribed poses (position and orientation). These linkages are also known under the name “Bennett linkage”. By that time, it was already known that three poses in general position define a unique Bennett linkage. Other synthesis methods were available [2–5]. With hindsight, the most important novelty of [1] was the characterization of the coupler motions of Bennett linkages as conics in the Study quadric model of the direct Euclidean displacement group SE(3). Depending on the precise concept of “Bennett linkages”, it might be necessary to exclude certain conics, see [6]. This interpretation allows to immediately construct the coupler motion from the three given poses—even before the linkage itself is determined.

In this article, we generalize the method of [1] to the four-pose synthesis of special kinematic loops of six revolute joints (6R linkages). Our main tool is the recently developed technique of factoring rational motions [7], or more precisely, motion polynomials. This allows to decompose a rational parametrized equation in Study parameters into the product of linear factors. Each such factorization determines an open kinematic chain. Suitable combinations of open chains produce closed loop linkages whose coupler follows the prescribed rational motion. If we combine this factorization of rational motions with rational interpolation techniques on a quadric, we obtain a framework for mechanism synthesis. It requires the completion of three steps:

− Interpolate \( n \) given poses by a rational motion of degree \( d \), parametrized by a motion polynomial \( C(t) \).
− Factor the motion polynomial \( C(t) \) and combine different factorizations to form a closed loop linkage.
− Pick a suitable linkage from the resulting family of solutions.

The mentioned synthesis of Bennett linkages corresponds to \( n = 3 \) poses and motion polynomials of degree \( d = 2 \). The motion polynomial \( C(t) \) admits two factorizations in two linear factors, each of which determines one fixed and one moving axis. These axes can be determined by geometric arguments as in [1] or by a straightforward algorithmic procedure (see Algorithm 2 below). The latter method extends to motions of higher degree. An important aspect of the suggested synthesis procedure is the fact that the mechanism is constructed from its coupler motion. Thus, optimal synthesis is not restricted to linkage characteristics only. It is possible and advisable to optimize the coupler motion as well.

In this paper we are concerned with the case \( n = 4 \) and \( d = 3 \). The reason for this is that, at least in general, higher degree motion polynomials result in multi-looped spatial linkages which are probably too complicated for most engineering
follow the motion parametrized by \( q \) of non-zero primal part in such a way that composition of displacements corresponds to dual quaternion multiplication.

In order to get rid of the ambiguity in the representation of elements of SE(3), one can factor by the real numbers and just not by \( \{ \pm 1 \} \). In this way, we arrive at the real projective space \( P^7 \). The unit norm condition reduces to the non-vanishing of the primal part and the vanishing of the dual part. The latter, a homogeneous quadratic form, defines the so-called Study quadric \( \mathcal{S} \), the former defines the exceptional three-space \( E \). The image of \( \text{SE}(3) \) under this kinematic map is \( \mathcal{S} \setminus E \). Homogeneous coordinates in \( P^7 \) are called Study parameters. In this context, a motion is a curve on \( \mathcal{S} \), possibly with isolated points in \( E \). We are interested in rational motions. These are defined by polynomial parametrizations in Study parameters or, equivalently, by motion polynomials.

Essentially, a motion polynomial is the parametrized equation of a rational motion in dual quaternions. More formally, we have:

**Definition 1.** A left polynomial \( C \) with dual quaternion coefficients is called a motion polynomial if \( C \bar{C} \) is a real polynomial, if its leading coefficient equals 1, and \( \deg C \) is greater than zero.

The condition on the leading coefficient is merely a technicality and constitutes no loss of generality. We denote the set of all left polynomials in the indeterminate \( t \) and with dual quaternion coefficients by \( \mathbb{DH}[t] \). The non-commutativity of dual quaternion multiplication entails some subtleties. To begin with, in the multiplication of two polynomials in \( \mathbb{DH}[t] \) the indeterminate \( t \) commutes, by definition, with all coefficients. This implies that there is a difference between left polynomials, with coefficients written to the left of the indeterminate, and right polynomials where coefficients are on the right. We consistently use left polynomials but often omit the additional word “left”. The sequence of polynomial multiplication matters and must strictly be obeyed in the algorithms we present below.

A dual quaternion \( h \) is called a rotation quaternion if it describes a rotation and a translation quaternion, if it describes a translation. Rotation quaternions are characterized by orthogonality of primal and dual part (Study condition), vanishing dual scalar part and non vanishing primal vector part. Translation quaternions are characterized by the Study condition, vanishing dual scalar part, vanishing primal vector part and non-vanishing primal scalar part. In this text, we will usually not encounter translation quaternions as we describe numeric procedures that produce translation quaternions with negligible probability.

**3 Factorization of motion polynomials**

A polynomial with real coefficients admits an essentially unique factorization into linear factors over the complex numbers. The situation for polynomials with dual quaternion coefficients is fundamentally different. Because of the non-commutative multiplication, a possible factorization is, in general, not unique. The defining conditions for motion polynomials (Definition 1) guarantees the existence of several special factorizations, at least in generic cases.

**Theorem 1** ([7]). For a generic motion polynomial \( C \in \mathbb{DH}[t] \) of degree \( n \) there exists \( n! \) different factorizations \( C(t) = (t - h_1) \cdots (t - h_n) \) with rotation quaternions \( h_1, \ldots, h_n \).

Here, the term “generic” refers to the non-vanishing of the primal part of \( C(t) \). Usually, zeros of primal \( C \) turn some of the rotation quaternions into translation quaternions. Motion polynomials that admit no factorization do exist. Although their precise characterization is not known, it is already clear that they are too special to be of real relevance in synthesis procedures. The total number \( n! \) of factorizations comes from permutations of the conjugate complex root pairs of the norm polynomial \( \bar{C}C \) of degree \( 2n \) (compare Algorithm 2 below). If some of these root pairs coincide, the actual number of factorizations may be less. Again, this non-generic case might be relevant to some special synthesis problems but not in this article.

Any factorization \( (t - h_1) \cdots (t - h_n) \) of a motion polynomial \( C \) encodes an open \( nR \)-chain whose end-effector can follow the motion parametrized by \( C(t) \) with \( t \in \mathbb{R} \cup \{ \infty \} \). The rotation quaternions \( h_1, \ldots, h_n \) give its axes in the zero
position \(C(\infty) = 1\). Different factorizations give rise to different open chains that can be combined to form closed loop linkages. Using factorization of motion polynomials, linkage synthesis is turned into a rational interpolation problem on the Study quadric. Rational interpolation on quadrics is a well-known topic in computer aided design [10, 11]. Relevant for our purposes are only low degree and low dimensional cases, whose geometry is well-understood.

A constructive proof of Theorem 1 can be found in [7]. Here, we describe the corresponding factorization algorithm. We denote the degree of \(C\) by \(\deg(C)\) and by \(\text{lc}(C)\) its leading coefficient. We use three auxiliary procedures:

- For a non-negative real polynomial \(P\), \(\text{FACTORS}(P)\) returns a list \([M_1, \ldots, M_n]\) of quadratic polynomials such that \(P = M_1 \cdots M_n\) and each factor \(M_i\) has at most a single real root. This list is unique up to permutation.
- For \(A, B \in \mathbb{D}[t]\) with \(\text{lc}(B) = 1\), \(Q = \text{QUO}(A, B)\) and \(R = \text{REM}(A, B)\) are the unique polynomials in \(\mathbb{D}[t]\) such that \(A = QB + R\) (quotient and remainder of polynomial division). For their computation, a single procedure \(QR(A, B)\) can be used. It is shown in Algorithm 1.

The function \(\text{RMUL}(D, P)\) right multiplies all left polynomials in a list \(D\) with the left polynomial \(P\), the function \(\text{APPEND}(R, D)\) appends a polynomial to a list of polynomials.

The computation of quotient and remainder works over arbitrary rings and in both, exact or floating point arithmetic. This is important because, in contrast to [7], we have to deal with non-rational input data or with motion polynomials \(C\) whose norm polynomial \(\overline{C}\) has no factorization over the rational numbers.

Algorithm 2 returns a list of all factorizations of a given motion polynomial \(C\). Our recursive implementation might not be the most efficient. For example the call of \(\text{FACTORS}(\overline{C})\) for each recursive call could be avoided because it produces only a subset of the call at the previous recursion level. Nonetheless, Algorithm 2 is easy to implement and fast for reasonably

---

**Algorithm 1 Quotient and remainder of polynomial right division**

```plaintext
procedure QR(A, B)  // Quotient and remainder of A/B; returns Q and R such that A = QB + R
Require: \(\text{lc}(B) = 1\)
Q ← 0
R ← A
while \(\deg(R) \geq \deg(B)\) do
  l ← \(\text{lc}(R)\)
  Q ← Q + l\(^{\deg(R) - \deg(B)}\)
  R ← R - Bl\(^{\deg(R) - \deg(B)}\)
end while
return Q, R
```

**Algorithm 2 Factorization of motion polynomials**

```plaintext
procedure FAC(C)  // All factorizations of C
Require: \(\text{lc}(C) = 1, C \in \mathbb{R}[t]\)
C ← \(\text{deg}(C)\)
if C = 0 then
  return C
end if
R ← []  // initialize empty list
F ← FACTORS(CC)
for i ← 1, n do
  L ← REM(C, M_i)
  if primal \(\text{lc}(L) = 0\) then
    ERROR!  // \(\text{lc}(L)\) not invertible; factorization fails
  end if
  h ← \(-\text{lc}(L)^{-1}L(0)\)
  C' ← QUO(C, t - h)
  D ← [FAC(C')]
  D ← RMUL(D, t - h)
  R ← APPEND(R, D)
end for
return R
```

---
small polynomial degrees. The pseudocode of Algorithm 2 shows that every factor \((t - h_i)\) in a particular factorization of \(C\) can be associated with a quadratic factor \(M_i\) of \(\mathcal{C}\). The complete factorization of \(C\) corresponds to a permutation of the factors of \(\mathcal{C}\). Moreover, we see that the factorization fails if the linear remainder polynomial has a non-invertible leading coefficient. In [7] we proved that this never happens in generic cases. We will ignore this possibility in the present investigation.

4 Cubic interpolation on the Study quadric

In Section 3 we saw how to factor a motion polynomial. Here, we discuss the construction of such polynomials to four given points \(p_1, p_2, p_3, p_4\) on the Study quadric \(\mathcal{S}\). More precisely, we want to find a cubic motion polynomial \(C(t)\) such that there exist values \(t_1, t_2, t_3, t_4 \in \mathbb{R} \cup \{\infty\}\) with \(C(t_i) \cong p_i\) for \(i \in \{1, 2, 3, 4\}\). Here, the symbol “\(\cong\)” means equality in projective sense, that is, up to multiplication with a non-zero real scalar.

Since the Study parameters are homogeneous, this can be seen as a particular instance of rational interpolation problem on quadrics [10, 11]. We are not interested in all aspect of this topic. In particular, the restriction to interpolants of degree \(n = 3\) is a considerable simplification. The case \(n = 2\) (interpolation by conics) is even simpler and there is no need to address it here. It has been extensively treated in [1, 6, 12].

We will study in more detail the interpolation of four points \(p_1, p_2, p_3, p_4\) on the Study quadric \(\mathcal{S} \subset \mathbb{P}^3\) by a rational cubic interpolant \(C(t)\) contained in \(\mathcal{S}\). Doing so, we will always assume that the four points \(p_1, p_2, p_3, p_4\) have a projective span \(P\) of dimension three. In order to exclude spherical or planar motions and particularities of translational joints, we assume that \(P\) is not contained in \(\mathcal{S}\) and does not intersect the exceptional generator \(E\). This last assumption is not really necessary but simplifies our exposition. Finally, it is helpful to set \(p_1 = 1\). This entails \(t_1 = \infty\) and can always be achieved by a suitable choice of coordinates.

Several methods for solving the interpolation problem are conceivable. We describe an approach that fits well to our dual quaternion based setting.

4.1 Existence of solutions

It is possible that no solution exists. This happens if the intersection \(Q = P \cap \mathcal{S}\) contains no real twisted cubics. Since \(Q\) is a real quadric in a projective three space, a simple criterion for the existence of cubics on \(Q\) is the existence of straight lines on \(Q\). We can test this by computing the straight lines on \(Q\) through \(p_1 = 1\). Any point \(k\) on a straight line \(L \subset \mathcal{S}\) through 1 is generically a rotation quaternion or, in special cases, a translation quaternion. In our description we ignore the latter possibility. Rotation quaternions have a purely vectorial dual part. Different points on the same line \(L\) correspond to different rotations about the same axes. The involutory rotations (half-turns) are characterized by a purely vectorial primal part.

In order to compute the half-turn quaternions in \(P\), we set \(k = x_1 p_1 + x_2 p_2 + x_3 p_3 + x_4 p_4\). The two half-turn conditions require the vanishing of the scalar part of \(k\). This gives two linear conditions, the Study quadric condition gives one quadratic conditions. Thus, we end up with a system of a quadratic and two linear equations for the homogeneous unknowns \(x_1, x_2, x_3, x_4\). It is easy to compute its up to two real solutions. In case of no real solution, the interpolation problem cannot be solved. The case of two real solutions is dealt with in the next section. The non-generic case of one solution is amenable to synthesis in the line of this article but, for reasons of compact presentation, will be ignored in the following.

4.2 Computing all solutions

We did not only compute the two half-turns \(k_1, k_2 \in K = P \cap S\) in order to test existence of solutions. We can also use them for actually computing interpolants. The quadric \(Q\) is doubly ruled, the span \(K_1\) of 1 and \(k_1\) belongs to one family of rulings and the span \(K_2\) of 1 and \(k_2\) to the other. Our aim is to find a twisted cubic interpolating \(p_1 = 1, p_2, p_3, p_4\). It is well-known that each such cubic intersects every member of one family of rulings in one point and each member of the other family in two points [13, p. 301]. Accordingly, we have to deal with two completely symmetric situations. We describe the case where the cubic intersects \(K_1\) twice and \(K_2\) once.

The second intersection point \(p_3\) of \(K_1\) and the cubic can be written as \(p_3 = \lambda - k_1\) with \(\lambda \in \mathbb{R} \cup \{\infty\}\). We use \(\lambda\) to parametrize all solution cubics in this family. The key observation is now that any rational cubic parametrization of the sought curve is induced, via the rulings in the family of \(K_2\), by a linear rational parametrization of any ruling in the family of \(K_1\). Hence, we compute parameter values \(t_i\) such that \(p_i' \equiv t_i - k_1\) is the intersection point of a ruling through \(p_i\) with \(K_1\) for \(i \in \{2, 3, 4\}\). The values \(t_2, t_3, t_4\) will be the parameter values at which the cubic curve interpolates the given points on the Study quadric. This is necessarily so, up to admissible change of parameter via a fractional linear re-parametrization. For the computation of \(t_i\) we have to evaluate the linear condition \(q(t_i - k_1, p_i) = 0\) where \(q\) is the quadratic form associated to the Study quadric. The value \(t_i = \infty\) cannot occur because of \(p_1 \neq p_i, i \in \{2, 3, 4\}\). Setting \(t_i = \infty\), we end up with five points \(p_1 = 1, p_2, p_3, p_4, p_5\) and corresponding parameter values \(t_i = \infty, t_2, t_3, t_4, \lambda\), chosen in such a way that a cubic interpolant
exists. Now we can use a standard interpolation algorithm to find a cubic interpolant $C$ such that $C(t_i) \equiv p_i$ for $i \in \{1, \ldots, 4\}$ and $C(\lambda) \equiv p_5$. This interpolant still depends on the parameter $\lambda$. By construction, the polynomial $C$ is a motion polynomial.

5 Synthesis

By combining the results of Section 3 and 4.2 we are, under certain mild assumptions, able to interpolate four given points $p_1 = 1, p_2, p_3, p_4 \in \mathcal{S}$ by two one-parametric families of rational cubic curves, parametrized by motion polynomials $C_\lambda(t)$ and $D_\mu(t)$, and factor each of these cubic polynomials, in six different ways, as $(t - h_1)(t - h_2)(t - h_3)$ with rotation quaternions $h_1, h_2, h_3$. Each triple $(h_1, h_2, h_3)$ of rotation quaternions corresponds to an open 3R chain that can reach the given poses $p_1, p_2, p_3, p_4$. Suitably combining two such chains gives a candidate 6R linkage for our synthesis problem.

However, there are invalid combinations of open 3R chains that should be avoided. Recall that each factorization obtained by Algorithm 2 corresponds to an ordering of the set $\{M_1, M_2, M_3\}$ of quadratic factors of the norm polynomial $CC$. We must not combine factorizations with the same quadratic factor at the first or the last position, as this produces a dangling link attached to either the end-effector or the base. On the level of factorizations, this can be distinguished by identical factors on the right or on the left. Six among the 15 possible combinations are afflicted with this defect. In order to exclude invalid linkages a priori, it is necessary that the FACTORS($CC$) procedure returns a list of factors that is sorted with respect to some total ordering relation. Thus, a consistent ordering of factors in recursive calls of FACT(C) is guaranteed. In this case, Algorithm 2 returns a list $R = [R_1, \ldots, R_6]$ of six factorizations in which the index pairs $(1, 4)$, $(1, 5)$, $(1, 6)$, $(2, 3)$, $(2, 4)$, $(2, 6)$, $(3, 5)$, $(3, 6)$, and $(4, 5)$ represent valid combinations. Three of these pairs, $(1, 6)$, $(2, 4)$, and $(3, 5)$, give rise to “general angle symmetric linkages” and six give rise to linkages of type “Waldron’s double Bennett hybrid” [7, 14].

At this point, we are left with two one-parametric families of six factorizations, each giving rise to nine candidate linkage. We should thus provide some method to pick “good” or at least “suitable” linkages from this variety. Before embarking on this, it must be said that a mathematical optimization on either of the families of solutions is problematic. In the description of Section 4.2, one family is parametrized by $\lambda \in \mathbb{R} \cup \{\infty\}$. Since the actual computation of a solution requires the factorization of a real polynomial of degree six and a subsequent polynomial division, differentiation with respect to $\lambda$ is difficult if not infeasible. However, since the solution space is only of dimension one, a straightforward line search is possible with reasonable effort.

5.1 Interpolation order

An important aspect in many synthesis problems is to guarantee a certain interpolation sequence. In the instance we describe, this is, unfortunately, not possible. From the construction in Section 4.2, it is obvious that the interpolation times $t_1 = \infty, t_2, t_3, t_4$ for the four given points are independent from the parameter $\lambda$ that determines the cubic interpolant. Within one family of solutions, the given poses are visited at exactly the same parameter times. The free parameters alone cannot be used to resolve a possible order defect. The only way to remedy this, is to change the input poses $p_1, p_2, p_3, p_4$.

5.2 Coupler motion

A specialty of the proposed procedure is the intermediate construction of possible end-effector motions from which the mechanism is synthesized. This adds an important ingredient to the problem of finding optimal linkages. Not only should certain linkage characteristics be satisfactory but the end-effector motion should be acceptable as well. This brings our topic in the vicinity of motion fairing [15, 16] and the construction of fair motions [17]. However, applicability of these established research topics to our problem should not be over-estimated. We do not recommend to waste our single free parameter on a fair coupler motion alone. It should also be used to produce satisfactory linkage properties. Thus, we use a suitable fairness functional only for eliminating linkages with bad coupler motion. The example in Section 6 below shows how this might work in applications. It is for this reason, that particular features or possible defects of the fairness functional, like dependence on chosen coordinate frames or disparity of translational and rotational units, are of minor relevance to us. In our example in Section 6 we use a very simple energy functional on the trajectories of certain feature points.

5.3 Linkage properties

Once candidate linkages with unfavorable end-effector motion have been eliminated, we can turn our attention to properties of the linkage itself. We suggest to compute the usual Denavit-Hartenberg parameters distances, angles, and offsets. From distances and offsets, it is possible to estimate the overall size of the linkage and exclude excessively large solutions. Small angles indicate a relatively “planar” spatial linkage, which might also be desirable in some cases.

While these are general considerations that also pertain to other types of linkages, we also suggest a quality measure that is tailored to linkages synthesized by factorization of cubic motion polynomials. It is related to the joint angles $\phi_1 = 0$ and $\phi_4$ at which the first and the last input pose are attained. In order to understand their relation to the interpolation parameter times $t_1 = \infty$ and $t_4$, we consider a rotation quaternion $h$ and the parametrization $t - h$ of its rotations. The rotation angle $\phi$
at parameter time $t \in \mathbb{R} \cup \{\infty\}$ is given by the formula

$$
\phi = 2 \arccot \frac{t - r}{\sqrt{4s - r^2}} \quad \text{where} \quad r = -h - \overline{h}, \ s = h\overline{h}.
$$

This formula has interesting consequences. The polynomial $M = t^2 + rt + s$ is the minimal polynomial of $h$—the unique monic left polynomial of degree two with $M(h) = 0$. In [7], we showed that if $t - h$ occurs in the factorization of a motion polynomial $C$, than $M$ is the corresponding factor of $\overline{C}$. By Algorithm 2, the rotation angles in two open 3R chains obtained through factorization of the same motion polynomial are permutations of each other. This implies existence of three joint pairs whose input-output relation is the identity. This was also observed in [14] and is the reason for the title of this paper.

Equation (1) implies that the rotation angle is strictly monotone and the linkage has full cycle mobility, at least generically. (These properties cannot be guaranteed if two consecutive revolute axes coincide. The rotation angle in this joint is then the sum of two rotation angles which may result in rocking joints.) This behavior of the joint angle functions is certainly a pleasant property of the synthesized linkages. It also suggests to use the angle increment in each joint between the first and last pose as a possible optimization target. In other words, the motion from $p_1$ to $p_4$ via $p_2$ and $p_3$ should be accomplished by small rotations in the six joints. Of course, this only makes sense, if the resulting linkage is not required to perform the full cycle end-effector motion. Since, by construction, the first pose is the identity and corresponds to zero rotation in all joints, we have to minimize the norm of the vector $(\omega_1, \omega_2, \omega_3)$ of joint angles associated with $p_4 = C(s_4)$. For this approach the following remarks hold true:

- The set of rotation angles is independent from the particular factorization of $C$. In fact, it can be computed from the factors of $\overline{C}$ alone and without running Algorithm 2.
- When using the maximum norm, the angle $\omega$ can be replaced by

$$
m(h_i) = \frac{|r|}{\sqrt{4s - r^2}}
$$

where $r = -h_i - \overline{h_i}, \ s = h_i\overline{h_i}$, and $h_i$ is the corresponding rotation quaternion. Again, $m(h_i)$ can be computed directly from the factors of the norm polynomial $\overline{C}$. We call it the factor’s angle characteristic. The assumption is that small rotations and sensible axis layout also produce good coupler motions in the range between $p_1$ and $p_4$.
- An optimal solution in this sense is associated to the motion polynomial $C$, not to a linkage. It is still necessary to make a choice from the nine linkages produced by the different factorizations of $C$.

6 An example

In this section we discuss different aspects of a particular synthesis problem. It is meant as an illustration of the general synthesis process and also as an example of how to incorporate the ideas of the preceding section in a synthesis pipeline. We want to synthesize an overconstrained 6R linkage to the four poses

$$
p_1 = 1,
p_2 = -0.575 + 0.374e + (0.598 - 0.194e)i + (0.397 + 0.310e)j + (0.393 + 0.529e)k,
p_3 = -0.312 + 0.939e + (0.903 + 0.116e)i + (0.189 + 0.219e)j + (0.225 + 0.653e)k,
p_4 = -0.688 + 0.808e + (0.719 + 0.678e)i - (0.098 + 0.686e)j + (0.012 + 0.086e)k.
$$

At first, we compute the two half-turn quaternions $k_1, k_2$ in the span of $p_1, p_2, p_3$, and $p_4$. They are

$$
k_1 = (0.546 - 0.115e)i + (0.583 - 0.174e)j + (0.602 + 0.273e)k,
k_2 = (0.810 + 1.575e)i + (0.252 - 3.011e)j + (0.530 - 0.973e)k.
$$

Their existence already implies that the interpolation problem admits solutions.

Now we have to construct interpolating cubic motion polynomials to the given poses. The theory predicts two one parametric families. The first step is the computation of possible parameter quadruples $(u_1, u_2, u_3, u_4)$ and $(v_1, v_2, v_3, v_4)$ for each family. Denote by $q$ the quadratic form associated to the Study quadric. The parameter values are computed from
the conditions \( q(p_i, u_i - k_1) = q(p_i, v_i - k_2) = 0 \) for \( i \in \{2, 3, 4\} \). For each input pose, this gives a system of seven linear equations with exactly one solution. In our example, these values are

\[
\begin{align*}
  u_1 &= \infty, & u_2 &= 0.660, & u_3 &= 0.368, & u_4 &= -0.034; \\
  v_1 &= \infty, & v_2 &= -0.294, & v_3 &= 0.304, & v_4 &= 0.575.
\end{align*}
\]

We added the parameter values \( u_1 \) and \( v_1 \) which, by construction, are infinity. The parameter space for one-parametric rational motions is the projective line. Hence, both quadruples visit the given poses in the order implied by their numbering. Neither of the two families suffers from an order defect.

Now we want to find rational cubics \( C(u) \) and \( D(v) \) such that \( C(u_i) \cong D(v_i) \cong p_i \) for \( i \in \{1, 2, 3, 4\} \). We only describe the computation of \( C(u) \). In order to avoid dealing with the parameter value \( u_1 = \infty \) we re-parametrize according to \( s(u) = u^{-1} \) and compute the cubic Lagrange interpolant \( \hat{C}(s) \) to the points \( w_i p_i, \ i \in \{1, 2, 3, 4\} \). The weights \( w_1, w_2, w_3, w_4 \) still have to be determined. For that purpose, we introduce a fifth point \( p_5 = \lambda - h_1 \) that depends on a free parameter \( \lambda \). The condition \( C(\lambda^{-1}) \cong p_5 \) gives us eight linear equations for \( w_1, w_2, w_3, w_4 \). By construction, they have a unique solution that still depends on \( \lambda \). In other words, we have found a one-parameter family of interpolating cubics \( C_\lambda(s) \). We re-substitute \( s = u^{-1} \) and multiply away the denominator \( u^3 \) to arrive at the input \( C_\lambda(u) \) for the next step. Similarly, we obtain a second family \( D_\mu(v) \) of solutions. Their formulas are too long to be displayed.

**Remark.** It is possible to extend the synthesis to geometric Hermite interpolation where we do not interpolate poses, but pose plus instantaneous velocity or even pose plus instantaneous velocity plus acceleration. The only necessary adaption is the replacement of Lagrange interpolation by Hermite interpolation. The calculations do not change much.

Let us now illustrate the importance of filtering out motions of poor fairness. Figure 1 displays a fair candidate motion on the left, a motion with bad fairness in the middle and the same bad motion in a different scaling on the right. The poses to be interpolated are indicated as well. It is obvious that the first motion is preferable. The fairness criterion we use in this example is

\[
F(\lambda) = \sum_{i=0}^{3} \int_{t_0}^{t_1} ||x_i'(t)|| \, dt
\]

where \( x_0(t), x_1(t), x_2(t) \) and \( x_3(t) \) are the trajectories of origin and the unit points on the three coordinate axes with respect to the motion defined by \( C_\lambda \). In other words, we measure fairness as sum of arc-lengths of certain trajectories. The trajectory of the origin is displayed in Figure 1. The arc-length as a measure of fairness is not very popular because of its tendency to straighten curve segments. In our case, it seems to be an acceptable choice because the single free parameter \( \lambda \) is not enough to “over-optimization” towards a kinky appearance.

We do not pick the optimal solution of Figure 1. Instead, we use (3) to eliminate solutions of bad fairness. In order to do so, we compute the fairness measures for a reasonable discretization of the solution space and only accept good solutions. In our example, we considered solutions in the first quartile as acceptable. This gives the two ranges \([27.609, 28.629]\) and \([26.917, 72.816]\) for the fairness of motions in the discretized sets \( \{C_{\lambda_0}, \ldots, C_{\lambda_n}\} \) and \( \{D_{\mu_0}, \ldots, D_{\mu_n}\} \), respectively. Since the optimal motion quality is comparable in both sets, we subject both to further scrutiny but restrict our description to the first set.

For each of the remaining motion polynomials \( C_\lambda \), we compute quadratic irreducible polynomials \( M_{i1}, M_{i2}, M_{i3} \in \mathbb{R}[t] \) such that the norm polynomial \( C_\lambda C_{\lambda_0} \) equals \( M_{i1} M_{i2} M_{i3} \). To each quadratic factor \( M_{i,j} = t^2 + r_{i,j} t + s_{i,j} \) we associate its...
angle characteristic, given by (2). It is our aim to keep the total rotation in the linkage small. For that purpose we pick the curve $C_\lambda$ and $D_\mu$ as respective minimizers of their maximal angle characteristics:

$$C(u) = 0.050 - 0.065\varepsilon - (0.055 + 0.052\varepsilon)i + (0.010 + 0.048\varepsilon)j + (0.002 - 0.008\varepsilon)k +$$

$$(-0.104 - 0.063\varepsilon + (0.035 + 0.045\varepsilon)i + (0.064 - 0.160\varepsilon)j + (0.075 - 0.054\varepsilon)k)\tau +$$

$$(-0.242 - 0.000\varepsilon - (0.321 - 0.177\varepsilon)i - (0.381 + 0.071\varepsilon)j - (0.377 + 0.247\varepsilon)k)i^2 + \tau^3,$$

$$D(v) = 0.260 - 0.336\varepsilon - (0.411 - 0.112\varepsilon)i - (0.208 + 0.264\varepsilon)j - (0.205 + 0.384\varepsilon)k +$$

$$(-0.884 + 0.115\varepsilon + (0.511 - 0.380\varepsilon)i + (0.559 + 0.389\varepsilon)j + (0.533 + 0.525\varepsilon)k)\tau +$$

$$(0.870 - 0.000\varepsilon - (0.374 + 0.363\varepsilon)i - (0.244 - 0.815\varepsilon)j - (0.320 - 0.162\varepsilon)k)\tau^2 + \tau^3.$$

The fairness values are $F(\lambda) = 28.629$ and $F(\mu) = 36.298$, respectively; the maximal angle characteristics are 1.268 and 2.041, respectively. The solution $C(u)$ is preferable in both respects. Hence, we discard $D(v)$.

Finally, we have to decide upon a suitable linkage among the nine linkages with coupler motion $C(u)$. We do this by minimizing the sum of distances in the linkage. The solution linkage and the motion of the end-effector coordinate system are visualized in Figure 2.

7 Synopsis

We presented a synthesis procedure for a certain type of overconstrained spatial 6R linkages that can visit four given poses. To the best of our knowledge, this is the first reasonable general and flexible synthesis method for 6R linkages. In lack of a concrete application, some aspects of our description necessarily remained rather general. We trust, however, that the example in Section 6 contains sufficient information and ideas to apply the suggested procedure to engineering tasks.
Our scheme obviously extend to more input poses and higher degree motion polynomials, however at the cost of increasing engineering complexity. Future research will focus on factorization of higher degree coupler motions with coinciding consecutive axes, thus keeping small the number of joints while allowing for more input poses. Another line of research is the synthesis of linkages whose coupler motion satisfies certain incidence constraints like point on line or point on sphere. This amounts to finding cubic curves on constraint varieties in the Study quadric.

Acknowledgments
This work was supported by the Austrian Science Fund (FWF): P 23831-N13.

References
[1] Brunnthaler, K., Schröcker, H.-P., and Husty, M., 2005. “A new method for the synthesis of Bennett mechanisms”. In Proceedings of CK 2005, International Workshop on Computational Kinematics.
[2] Veldkamp, G. R., 1967. “Canonical systems and instantaneous invariants in spatial kinematics”. J. Mechanisms, 3, pp. 329–388.
[3] Suh, C. H., 1969. “On the duality in the existence of R-R links for three positions”. ASME J. Ind., 91(B), pp. 129–134.
[4] Tsai, L. W., and Roth, B., 1973. “A note on the design of revolute-revolute cranks”. Mech. Mach. Theory, 8, pp. 23–31.
[5] Perez, A., and McCarthy, J. M., 2004. “Dimensional synthesis of Bennett linkages”. ASME J. Mech. Design, 126(3), pp. 425–435.
[6] Hamann, M., 2011. “Line-symmetric motions with respect to reguli”. Mech. Mach. Theory, 46(7), pp. 960–974.
[7] Hegedüs, G., Schicho, J., and Schröcker, H.-P., 2013. “Factorization of rational curves in the Study quadric and revolute linkages”. Mech. Mach. Theory, 69(1), pp. 142–152.
[8] Perez, A., and McCarthy, J. M., 2003. “Dual quaternion synthesis of constrained robotic systems”. J. Mechanical Design, 126(3), pp. 425–435.
[9] McCarthy, J. M., and Soh, G. S., 2011. Geometric Design of Linkages, 2 ed., Vol. 11 of Interdisciplinary Applied Mathematics. Springer, New York.
[10] Gfrerrer, A., 1999. “Rational interpolation on a hypersphere”. Comput. Aided Geom. Design, 16(1), pp. 21–37.
[11] Gfrerrer, A., 2000. “On the construction of rational curves on hyperquadrics”. Grazer Math. Ber., 340, pp. 1–69.
[12] Schröcker, H.-P., and Jüttler, B., 2009. “Motion interpolation with Bennett biarcs”. In Proceedings of Computational Kinematics (CK 2009), A. Kecskeméthy and A. Müller, eds. Springer, pp. 101–108.
[13] Salmon, G., 1882. A treatise on the analytic geometry of three dimensions, 4 ed. Hodges, Figgis & Co., Dublin.
[14] Li, Z., and Schicho, J., 2013. “Classification of angle-symmetric 6R linkages”. Mech. Mach. Theory, 70, pp. 372–379.
[15] Fang, Y. C., Hsieh, C. C., Kim, M. J., Chang, J. J., and Woo, T. C., 1998. “Real time motion fairing with unit quaternions”. CAD, 30(3), pp. 191–198.
[16] Hsieh, C.-C., and Chang, T.-Y., 2003. “Motion fairing using genetic algorithms”. CAD, 35(8), pp. 739–749.
[17] Hofer, M., Pottmann, H., and Ravani, B., 2004. “From curve design algorithms to the design of rigid body motions”. The Visual Computer, 20(5), pp. 279–297.