Kinematic geometry of spatial RSSR mechanisms

Mirja Rotzoll[a], Margaret H. Regan[b], Manfred L. Husty[c], M. John D. Hayes[a,*]

[a] *Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada*
[b] *Duke University, 120 Science Drive Physics 117, Durham, NC 27708, U.S.A.*
[c] *University of Innsbruck, Technikerstraße 13, 6020 Innsbruck, Austria*

Abstract

Two different novel methods to derive the input-output (IO) equation of arbitrary RSSR linkages are described. Both methods share some common elements, i.e., they use the standard Denavit-Hartenberg notation to first describe the linkage as an open kinematic chain, and Study's kinematic mapping to describe the displacement of the coordinate frame attached to the end-effector of the chain with respect to the relatively non-moving base frame. The kinematic closure equation is obtained in the seven-dimensional projective kinematic mapping image space by equating the eight Study soma coordinates to the identity array. Then two methods are successfully applied to eliminate the intermediate joint angle parameters leading to the degree four biquadratic implicit algebraic IO equation: a) the linear implicitisation algorithm, which can be applied after rearranging the closure equation such that the linkage can be viewed as two serial RS chains, and b) numerical elimination theory using pseudowitness sets. Both approaches lead to the same IO equation. The utility of this algebraic form of the IO equation is illustrated with three detailed application examples.

Keywords: RSSR linkage, Study soma coordinates, algebraic input-output equation, linear implicitisation algorithm.

1. Introduction

The RSSR mechanism has been investigated since 1955 [1], if not earlier. It has been broadly used in modern applications ranging from hinging to landing gear deployment systems so there has long been a need for design tools for synthesis and analysis. The earliest works considering mobility limits date from as early as 1969, see [2, 3, 4, 5]. Displacement and dynamic analysis of the RSSR dates from 1972, if not earlier [6, 7]. Optimal synthesis of RSSR linkages for various objectives can be traced to the early 1980s [8], but there is also modern interest, see [9] for example. Rigid body motion synthesis using Study's kinematic mapping [10] was elegantly developed for planar four-bar

*Corresponding Author
Email address: john.hayes@carleton.ca (M. John D. Hayes)

linkages in [11]. Motivated by this, a derivation algorithm that describes the linkage using Denavit Hartenberg (DH) parameters, projects the displacement transformation matrix into Study's kinematic image space, and manipulates the resulting equations via Gröbner bases to obtain the algebraic input-output (IO) equation for planar, spherical, and Bennett linkages has been investigated with results reported in the literature by the authors of this paper. A natural extension of this algorithm to general motion in three dimensional space is to apply it to another well-investigated spatial linkage, the RSSR, which will be the main focus of this paper. In addition, the results obtained using the polynomial elimination method [12] are supported by a numerical method [13] leading to an identical algebraic IO equation, as well as a verification of the equation using an animated example linkage that was created in the GeoGebra software.

It is important to note that we make no claims regarding the relative ease or difficulty of the method presented in this paper for deriving the RSSR algebraic IO equation compared to any existing method, we simply claim that it is different. However, to underscore the utility of this form of the algebraic RSSR IO equation as the cornerstone for development of powerful novel mechanical design tools for synthesis and analysis, three detailed example applications are presented: continuous approximate synthesis for function generation minimising the design and structural errors; mobility limits; and extreme values of angular velocity and acceleration.

The RSSR linkage consists of two revolute (R) and two spherical (S) joints and following the Kutzbach criterion, possesses 2 degrees of freedom (dof). However, one dof that does not influence the IO equation corresponds to the rotation of the coupler link between the two spherical joints about its own longitudinal axis. This so-called idle dof can have a positive effect on the durability of the linkage in engineering applications, as it helps to evenly wear the S joints. Generally, the IO equation of the RSSR is much more involved compared to the planar, and spherical ones, as in addition to the link lengths between the four joints, the linkage further possesses three additional design parameters between the revolute joints, i.e., two link offsets and a link twist. Previous trigonometric derivations of the RSSR IO equation are available, for example, in [1, 14, 15]. Hartenberg and Denavit's derivation of the IO equation [14] uses their well-known parameters and trigonometric relations, while the derivation in [15] leads to an equation that resembles a more complex version of the Freudenstein equation [16]. This is not entirely surprising given that the planar four-bar is a special case of the RSSR linkage.

2. Denavit-Hartenberg (DH) Parametrisation

The literature contains many variations of the original Denavit-Hartenberg (DH) coordinate system and parameter assignment convention. For example, subtly different coordinate frame attachment rules and parameter definitions have been devised for mechanical system calibration, dynamic analysis, accounting for misalignment of joint axis directions, etc., see [17, 18, 19, 20] for several

different examples. Therefore, it is important to precisely define the convention used in this work to avoid confusion and misinterpretation since the corresponding coordinate transformations are all different from those of Denavit and Hartenberg.

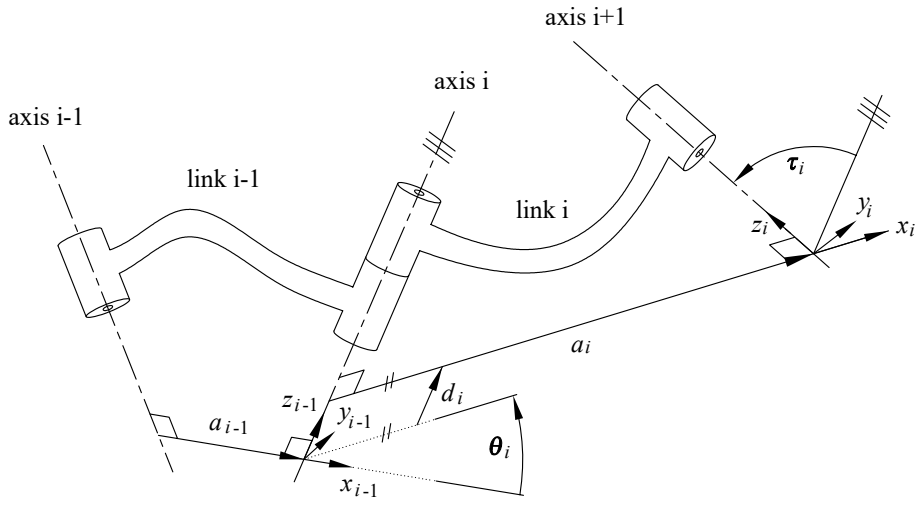


Figure 1: DH parameters in a general serial 3R kinematic chain.

To visualise the four DH parameters, consider two arbitrary sequential neighbouring links, $i-1$ and i. Two such links are illustrated, together with their DH coordinate systems and parameters, in Fig. 1. The DH parameters [21] are defined in the following way.

θ_i, joint angle: the angle from x_{i-1} to x_i measured about z_{i-1}.

d_i, link offset: the distance from x_{i-1} to x_i measured along z_{i-1}.

τ_i, link twist: the angle from z_{i-1} to z_i measured about x_i.

a_i, link length: the directed distance from z_{i-1} to z_i measured along x_i.

Each of the two S joints of the RSSR can be modelled as three R joints whose rotation axes are mutually orthogonal and intersect at the sphere centre. Hence, eight coordinate frames are attached to the linkage. The chosen coordinate systems are illustrated in Fig. 2 and the corresponding DH parameters are to be found in Table 1. Note that the only link twist that is a design parameter is τ_8. The twists between the three mutually orthogonal R joint axes comprising the S joints are $\pm\pi$. We arbitrarily use the positive value, as the sign has no impact on the resulting algebraic IO equation.

In the remainder of this paper, the tangent half angle substitutions for the angle parameters $v_i = \tan(\theta_i/2)$ and $\alpha_i = \tan(\tau_i/2)$ will be used in order to

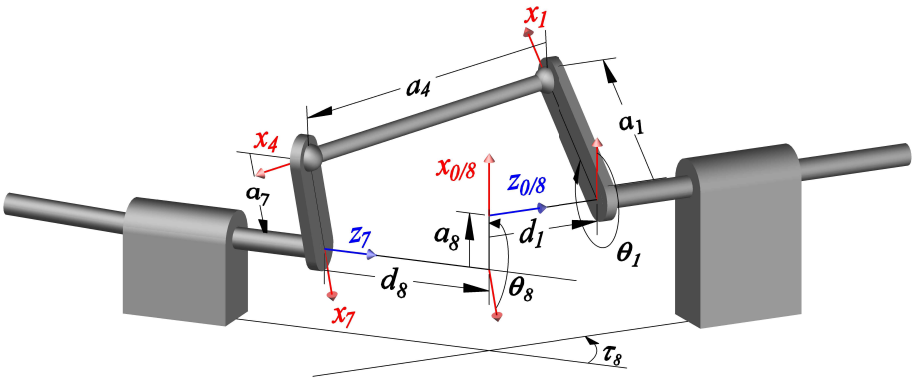


Figure 2: An arbitrary RSSR mechanism.

Table 1: DH parameters for the RSSR mechanism.

joint axis i	joint angle θ_i	link offset d_i	link length a_i	link twist τ_i
1	θ_1	d_1	a_1	0
2	θ_2	0	0	$\pi/2$
3	θ_3	0	0	$\pi/2$
4	θ_4	0	a_4	0
5	θ_5	0	0	$\pi/2$
6	θ_6	0	0	$\pi/2$
7	θ_7	0	a_7	0
8	θ_8	d_8	a_8	τ_8

algebraise the transformations. This implies that

$$\cos\theta_i = \frac{1-v_i^2}{1+v_i^2}, \quad \sin\theta_i = \frac{2v_i}{1+v_i^2}, \tag{1}$$

$$\cos\tau_i = \frac{1-\alpha_i^2}{1+\alpha_i^2}, \quad \sin\tau_i = \frac{2\alpha_i}{1+\alpha_i^2}. \tag{2}$$

We begin with a serial RSSR kinematic chain and determine the forward kinematics following [21]. The required multiplication of the individual DH transformation matrices from one coordinate frame to another yields the overall homogeneous transformation matrix that describes the relationship between the first and last coordinate frames. To close the kinematic chain, we want the first and last coordinate systems to align in both their orientation and origin. Algebraically, this is specified using the kinematic closure equation, where the

overall transformation equates to the identity [21]

$$\prod_{i=1}^{8} {}_{i}^{i-1}\mathbf{T} = \mathbf{I}. \tag{3}$$

The elements of this algebraic DH transformation matrix are then directly mapped into Study's kinematic image space where the constraint manifold could be analysed as it has already been successfully demonstrated for the planar 4R, spherical 4R, and Bennett linkage by the authors. However, applying Gröbner bases or other elimination methods to the eight Study soma coordinates to symbolically obtain the IO equation for the RSSR linkage is computationally too demanding for an algebraic geometry approach. While very computationally demanding, a numerical approach that uses the forward kinematics of the serial RSSR chain mapped to the eight soma coordinates, described in Section 5, using pseudowitness sets leads directly to the desired IO equation. Still, there are algebraic approaches.

A well known algebraic geometry approach to obtain an expression for the forward and inverse kinematics of a serial kinematic chain is to split it into two subchains, thereby conceptually splitting the closure equation in two by multiplying both sides by the inverses of half of the DH transformations. In the case of the RSSR, the closure equation becomes

$${}_{1}^{0}\mathbf{T}\ {}_{2}^{1}\mathbf{T}\ {}_{3}^{2}\mathbf{T}\ {}_{4}^{3}\mathbf{T} = \mathbf{I}\ {}_{8}^{7}\mathbf{T}^{-1}\ {}_{7}^{6}\mathbf{T}^{-1}\ {}_{6}^{5}\mathbf{T}^{-1}\ {}_{5}^{4}\mathbf{T}^{-1}. \tag{4}$$

This step essentially divides the linkage into two serial chains joined at the 4th coordinate frame located in the second S joint, i.e., one serial chain between the coordinate frames 0 and 4, and one serial chain between the coordinate frames 4 and 8, which correspond to the expressions on the left and right sides of Eq. (4), respectively, which we call the left RS and right RS dyads. Eq. (4) will be used in Section 4 to obtain the algebraic IO equation by projecting it to the image space. However, before we proceed we will briefly recall Study's kinematic mapping [10].

3. Study's Kinematic Mapping

The homogeneous transformation matrices in Eqs. (3) and (4) represent a subgroup of the group of spatial Euclidean displacements, $SE(3)$, with respect to a relatively non-moving coordinate frame. There are several possibilities to parameterise this rigid body displacement group, one of them being the kinematic mapping that was originally formulated by Eduard Study and reported in an appendix of his book "Geometrie der Dynamen" [10] in 1903. It defines every distinct Euclidean displacement as a distinct point on a six-dimensional quadric hyper-surface in a seven-dimensional projective space $\mathbb{P}^7$ now known as the Study quadric, S_6^2. A point on S_6^2 consists of eight homogeneous coordinates, not all zero, $[x_0 : x_1 : x_2 : x_3 : y_0 : y_1 : y_2 : y_3]$ which Study called a "soma", a

Greek word meaning "body". The hyper-surface is a seven-dimensional bilinear hyper-quadratic equation given by

$$x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 = 0, \tag{5}$$

excluding the *exceptional generator*, which we call A_∞, where $x_0 = x_1 = x_2 = x_3 = 0$, having the parametric representation

$$[0 : 0 : 0 : 0 : y_0 : y_1 : y_2 : y_3].$$

A_∞ does not represent any real displacement, but it nonetheless plays an important role as a generator space. For a soma to represent a real displacement in $SE(3)$, it must satisfy two conditions: the first being Eq. (5); the second being the inequality

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 \quad \neq \quad 0. \tag{6}$$

Eq. (5) contains only bilinear cross terms. This implies that the quadric has been rotated out of its standard position, or normal form. It is straightforward to diagonalise the quadratic form of Eq. (5) which reveals that this six-dimensional quadric in $\mathbb{P}^7$ has the normal form

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 - y_0^2 - y_1^2 - y_2^2 - y_3^2 = 0, \tag{7}$$

which is analogous to the Plücker quadric, P_4^2, of line geometry. The normal form of S_6^2 shows that it is a six-dimensional hyperboloid of one sheet doubly-ruled by special 3-space generators in two opposite reguli, which we call A-planes and B-planes, after [22].

It can be shown that lines on S_6^2 represent either a one parameter set of translations or rotations. The lines which contain the 1×8 identity array $[1 : 0 : 0 : \ldots : 0]$, which Study called the "protosoma", are either the one parameter rotation or translation subgroups. The exceptional generator A_∞ is an A-plane. In general, two different A-planes do not intersect, nor do two different B-planes, but there are exceptions [23]. An A-plane corresponds to $SO(3)$ if it contains the identity and its intersection with A_∞ is the empty set, and to $SE(2)$ if it contains the identity and intersects A_∞ in a line. These two types of A-planes intersect each other in lines on S_6^2. Each of these lines represent rotations about the line orthogonal to the plane of the planar displacement and through the centre point of the spherical displacement [23, 24]. The only B-planes that intersect A_∞ correspond to the subgroup of all translations, while in general the intersection of an A-plane and a B-plane is either a point, or a two dimensional plane [25].

Given a homogeneous transformation matrix $\mathbf{T}$ whose 3×3 rotation submatrix elements are denoted as $\mathbf{A} = (a_{ij})$ with $i, j \in \{1, 2, 3\}$ and whose translation vector elements are denoted as t_k with $k \in \{1, 2, 3\}$, then the corresponding Study soma coordinates, also known as Study parameters, are obtained in the following way. The homogeneous quadruple $x_0 : x_1 : x_2 : x_3$ can be obtained

from at least one of the following ratios:

$$\begin{aligned} x_0 : x_1 : x_2 : x_3 &= 1 + a_{11} + a_{22} + a_{33} : a_{32} - a_{23} : a_{13} - a_{31} : a_{21} - a_{12}; \\ &= a_{32} - a_{23} : 1 + a_{11} - a_{22} - a_{33} : a_{12} + a_{21} : a_{31} + a_{13}; \\ &= a_{13} - a_{31} : a_{12} + a_{21} : 1 - a_{11} + a_{22} - a_{33} : a_{23} + a_{32}; \\ &= a_{21} - a_{12} : a_{31} + a_{13} : a_{23} + a_{32} : 1 - a_{11} - a_{22} + a_{33}. \end{aligned} \tag{8}$$

The remaining four coordinates $y_0 : y_1 : y_2 : y_3$ are linear combinations of the x_i and t_i and are computed as

$$\begin{aligned} y_0 &= \tfrac{1}{2}(t_1x_1 + t_2x_2 + t_3x_3), & y_1 &= \tfrac{1}{2}(-t_1x_0 + t_3x_2 - t_2x_3), \\ y_2 &= \tfrac{1}{2}(-t_2x_0 - t_3x_1 + t_1x_3), & y_3 &= \tfrac{1}{2}(-t_3x_0 + t_2x_1 - t_1x_2). \end{aligned} \tag{9}$$

Study developed the method to compute the four x_i parameters directly from the 3×3 rotation submatrix $\mathbf{A}$ via one of the four sets of ratios expressed in Eq. (8). In general each of the four yield the same ratios. But in certain instances, for example when $\mathbf{A}$ describes a rotation through angle π, one or more of the four ratios in Eq. (8) result in $x_0 : x_1 : x_2 : x_3 = 0 : 0 : 0 : 0$, the exceptional generator. But for every rotation matrix $\mathbf{A}$ at least one of the four ratios does not result in four zeros. Study also showed that the mapping is bijective, meaning that for each point on S_6^2 there is one and only one Euclidean displacement represented by the homogeneous 4×4 transformation matrix $\mathbf{T}$:

$$\mathbf{T} = \frac{1}{\delta}\begin{bmatrix} x_0^2 + x_1^2 + x_2^2 + x_3^2 & 0 & 0 & 0 \\ 2(-x_0y_1 + x_1y_0 - x_2y_3 + x_3y_2) & x_0^2 + x_1^2 - x_2^2 - x_3^2 & 2(-x_0x_3 + x_1x_2) & 2(x_0x_2 + x_1x_3) \\ 2(-x_0y_2 + x_1y_3 + x_2y_0 - x_3y_1) & 2(x_0x_3 + x_1x_2) & x_0^2 - x_1^2 + x_2^2 - x_3^2 & 2(-x_0x_1 + x_2x_3) \\ 2(-x_0y_3 - x_1y_2 + x_2y_1 + x_3y_0) & 2(-x_0x_2 + x_1x_3) & 2(x_0x_1 + x_2x_3) & x_0^2 - x_1^2 - x_2^2 + x_3^2 \end{bmatrix}$$

where $\delta = x_0^2 + x_1^2 + x_2^2 + x_3^2$. The first column is the associated translation of the Euclidean displacement and the elements of the lower right 3×3 submatrix are the nine a_{ij} of the associated rotation matrix $\mathbf{A}$. Hence, the mechanical constraints imposed by the type of joints used in the kinematic chains of the RSSR are mapped onto Study's quadric. The result is a parametric representation in terms of Study soma coordinates of the constraint manifold.

The image of the overall DH transformation matrix $\mathbf{T}$ of the RSSR linkage, Eq. (3), in terms of Study parameters yields

$$\begin{aligned} x_0 &= 2v_1v_2v_3v_4v_5v_6v_7v_8 - 2v_1v_2v_3v_4v_5v_6 + ... + 2\alpha_8v_6v_8 + 2v_7v_8 - 2, \\ x_1 &= 2\alpha_8v_1v_2v_3v_4v_5v_6v_7v_8 - 2\alpha_8v_1v_2v_3v_4v_5v_6 + ... + 2\alpha_8v_7v_8 - 2\alpha_8, \\ x_2 &= -2\alpha_8v_1v_2v_3v_4v_5v_6v_7 - 2\alpha_8v_1v_2v_3v_4v_5v_6v_8 + ... - 2\alpha_8v_7 - 2\alpha_8v_8, \\ x_3 &= -2v_1v_2v_3v_4v_5v_6v_7 - 2v_1v_2v_3v_4v_5v_6v_8 + ... + 2\alpha_8v_6 - 2v_7 - 2v_8, \\ y_0 &= -a_1\alpha_8v_1v_2v_3v_4v_5v_6v_7v_8 + a_4\alpha_8v_1v_2v_3v_4v_5v_6v_7v_8 + ... - \alpha_8a_8, \\ y_1 &= a_1v_1v_2v_3v_4v_5v_6v_7v_8 - a_4v_1v_2v_3v_4v_5v_6v_7v_8 + ... + a_1 + a_4 + a_7 + a_8, \\ y_2 &= -\alpha_8d_1v_1v_2v_3v_4v_5v_6v_7v_8 - \alpha_8d_8v_1v_2v_3v_4v_5v_6v_7v_8 + ... + \alpha_8d_8, \\ y_3 &= -d_1\alpha_8v_1v_2v_3v_4v_5v_6v_7v_8 - d_8v_1v_2v_3v_4v_5v_6v_7v_8 + ... + d_1 + d_8. \end{aligned} \tag{10}$$

As these polynomials are extremely large, each containing 128 very large terms, only the beginning and end of the expressions sorted using graded lexicographic ordering with $v_1 > v_2 > \ldots > v_8$ are displayed here. These polynomials will be solved numerically in Section 5, but are otherwise too cumbersome to deal with using algebraic geometry and computer algebra software, such as Maple 2021. For this we require a different approach.

As mentioned earlier, one well known different approach involves conceptually splitting the RSSR into two serial RS chains. In this way, mapping the left hand side of Eq. (4), the left RS chain, into Study's kinematic image space yields eight significantly smaller polynomials

$$
\begin{aligned}
x_0 &= 4v_1v_2v_3v_4 - 4v_1v_3 - 4v_2v_3 - 4v_3v_4,\\
x_1 &= -4v_1v_2 + 4v_1v_4 + 4v_2v_4 + 4,\\
x_2 &= 4v_1v_2v_4 + 4v_1 + 4v_2 - 4v_4,\\
x_3 &= -4v_1v_2v_3 - 4v_1v_3v_4 - 4v_2v_3v_4 + 4v_3,\\
y_0 &= -2d_1v_1v_2v_3 - 2d_1v_1v_3v_4 - 2d_1v_2v_3v_4 + 2a_1v_1v_2 - 2a_4v_1v_2 - 2a_1v_1v_4\\
&\quad + 2a_4v_1v_4 + 2a_1v_2v_4 + 2a_4v_2v_4 + 2d_1v_3 + 2a_1 + 2a_4,\\
y_1 &= 2a_1v_1v_2v_3v_4 - 2a_4v_1v_2v_3v_4 + 2d_1v_1v_2v_4 - 2a_1v_1v_3 + 2a_4v_1v_3 + 2a_1v_2v_3\\
&\quad + 2a_4v_2v_3 + 2a_1v_3v_4 + 2a_4v_3v_4 + 2d_1v_1 + 2d_1v_2 - 2d_1v_4,\\
y_2 &= 2a_1v_1v_2v_3 + 2a_4v_1v_2v_3 + 2a_1v_1v_3v_4 + 2a_4v_1v_3v_4 - 2a_1v_2v_3v_4 + 2a_4v_2v_3v_4\\
&\quad + 2d_1v_1v_2 - 2d_1v_1v_4 - 2d_1v_2v_4 + 2a_1v_3 - 2a_4v_3 - 2d_1,\\
y_3 &= -2d_1v_1v_2v_3v_4 + 2a_1v_1v_2v_4 + 2a_4v_1v_2v_4 + 2d_1v_1v_3 + 2d_1v_2v_3 + 2d_1v_3v_4\\
&\quad + 2a_1v_1 + 2a_4v_1 - 2a_1v_2 + 2a_4v_2 + 2a_1v_4 - 2a_4v_4.
\end{aligned}
\tag{11}
$$

And finally, mapping the right hand side of Eq. (4), the right RS chain, into Study's kinematic image space yields eight additional smaller polynomials

$$
\begin{aligned}
x_0 &= 4v_5v_6v_7v_8 - 4v_5v_6 - 4\alpha_8v_5v_7 - 4\alpha_8v_5v_8 - 4v_6v_7 - 4v_6v_8 + 4\alpha_8v_7v_8 - 4\alpha_8,\\
x_1 &= -4\alpha_8v_5v_6v_7v_8 + 4\alpha_8v_5v_6 - 4v_5v_7 - 4v_5v_8 + 4\alpha_8v_6v_7 + 4\alpha_8v_6v_8 + 4v_7v_8 - 4,\\
x_2 &= 4\alpha_8v_5v_6v_7 + 4\alpha_8v_5v_6v_8 + 4v_5v_7v_8 + 4\alpha_8v_6v_7v_8 - 4v_5 - 4\alpha_8v_6 + 4v_7 + 4v_8,\\
x_3 &= 4v_5v_6v_7 + 4v_5v_6v_8 - 4\alpha_8v_5v_7v_8 + 4v_6v_7v_8 + 4\alpha_8v_5 - 4v_6 - 4\alpha_8v_7 - 4\alpha_8v_8,\\
y_0 &= -2a_7\alpha_8v_5v_6v_7v_8 + 2a_8\alpha_8v_5v_6v_7v_8 - 2d_8v_5v_6v_7 - 2d_8v_5v_6v_8 - 2\alpha_8d_8v_5v_7v_8\\
&\quad - 2d_8v_6v_7v_8 - 2a_7\alpha_8v_5v_6 - 2a_8\alpha_8v_5v_6 + 2a_7v_5v_7 + 2a_8v_5v_7 - 2a_7v_5v_8\\
&\quad + 2a_8v_5v_8 - 2a_7\alpha_8v_6v_7 - 2a_8\alpha_8v_6v_7 + 2a_7\alpha_8v_6v_8 - 2a_8\alpha_8v_6v_8 + 2a_7v_7v_8\\
&\quad - 2a_8v_7v_8 + 2\alpha_8d_8v_5 + 2d_8v_6 - 2\alpha_8d_8v_7 - 2\alpha_8d_8v_8 + 2a_7 + 2a_8,\\
y_1 &= -2a_7v_5v_6v_7v_8 + 2a_8v_5v_6v_7v_8 + 2\alpha_8d_8v_5v_6v_7 + 2\alpha_8d_8v_5v_6v_8 - 2d_8v_5v_7v_8\\
&\quad + 2\alpha_8d_8v_6v_7v_8 - 2a_7v_5v_6 - 2a_8v_5v_6 - 2a_7\alpha_8v_5v_7 - 2a_8\alpha_8v_5v_7 + 2a_7\alpha_8v_5v_8\\
&\quad - 2a_8\alpha_8v_5v_8 - 2a_7v_6v_7 - 2a_8v_6v_7 + 2a_7v_6v_8 - 2a_8v_6v_8 - 2a_7\alpha_8v_7v_8\\
&\quad + 2a_8\alpha_8v_7v_8 + 2d_8v_5 - 2\alpha_8d_8v_6 - 2d_8v_7 - 2d_8v_8 - 2a_7\alpha_8 - 2a_8\alpha_8,\\
y_2 &= 2\alpha_8d_8v_5v_6v_7v_8 - 2a_7v_5v_6v_7 - 2a_8v_5v_6v_7 + 2a_7v_5v_6v_8 - 2a_8v_5v_6v_8\\
&\quad - 2a_7\alpha_8v_5v_7v_8 + 2a_8\alpha_8v_5v_7v_8 + 2a_7v_6v_7v_8 - 2a_8v_6v_7v_8 - 2\alpha_8d_8v_5v_6
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
&- 2d_8v_5v_7 - 2d_8v_5v_8 - 2\alpha_8d_8v_6v_7 - 2\alpha_8d_8v_6v_8 + 2d_8v_7v_8 - 2a_7\alpha_8v_5 \\
&- 2a_8\alpha_8v_5 + 2a_7v_6 + 2a_8v_6 + 2a_7\alpha_8v_7 + 2a_8\alpha_8v_7 - 2a_7\alpha_8v_8 + 2a_8\alpha_8v_8 - 2d_8, \\
y_3 = {} & 2d_8v_5v_6v_7v_8 + 2a_7\alpha_8v_5v_6v_7 + 2a_8\alpha_8v_5v_6v_7 - 2a_7\alpha_8v_5v_6v_8 + 2a_8\alpha_8v_5v_6v_8 \\
&- 2a_7v_5v_7v_8 + 2a_8v_5v_7v_8 - 2a_7\alpha_8v_6v_7v_8 + 2a_8\alpha_8v_6v_7v_8 - 2d_8v_5v_6 \\
&+ 2\alpha_8d_8v_5v_7 + 2\alpha_8d_8v_5v_8 - 2d_8v_6v_7 - 2d_8v_6v_8 - 2\alpha_8d_8v_7v_8 - 2a_7v_5 - 2a_8v_5 \\
&- 2a_7\alpha_8v_6 - 2a_8\alpha_8v_6 + 2a_7v_7 + 2a_8v_7 - 2a_7v_8 + 2a_8v_8 + 2\alpha_8d_8.
\end{aligned}
$$

The polynomials of Eqs. (11) and (12) will be manipulated in Section 4 using the linear implicitisation algorithm [12] to reveal the algebraic RSSR IO equation.

4. Algebraic Geometry Approach

To obtain the RSSR algebraic IO equation, the parametric equations of the Study coordinates of Eqs. (11) and (12) need to be expressed implicitly as a single polynomial equation in the desired motion parameters v_1 and v_8 in the seven-dimensional kinematic mapping image space. This requires an algorithm that eliminates the unwanted motion parameters v_i where $i \in \{2, \ldots, 7\}$. One implicitisation algorithm that allows for the transformation from the explicit parametric Study representation into a set of implicit polynomial equations is known as the linear implicitisation algorithm. The resulting constraint equations are implicit polynomials that form an algebraic variety in $\mathbb{P}^7$ and can be manipulated with different tools to obtain the IO equation. A detailed description of the linear implicitisation algorithm, together with illustrative examples is to be found in [12, 26].

The two serial RS chains of the RSSR linkage consist of one revolute and one spherical joint each. Clearly, the S joint spherical displacements, $SO(3)$, are completely contained on sub-spaces of the Study quadric as there is no translation involved and thus, all four y_i Study coordinates are identically zero. In other words, the displacements constrained by the S joints form special A-planes on the Study quadric. Further, the R joint in the serial RS chain rotates the S joint in a planar displacement thereby moving this special A-plane on S_6^2. It is well known that a 3-space can be represented by the intersection of four hyperplanes in the kinematic image space. To determine the RSSR algebraic IO equation we must identify these hyperplanes, one set for each serial RS chain. To obtain their implicit equations the linear implicitisation algorithm will be employed. The main goal of the linear implicitisation algorithm is to find the minimal number of implicit equations that describe the mechanical constraints in the image space. It allows for the elimination of motion parameters which, in the case of the RSSR, correspond to the variables $v_2, v_3, \ldots, v_7$. On the other hand, the design parameters a_i, d_i and α_i are fixed values that depend on the chosen linkage. However, to obtain the implicit polynomials for the spherical special 3-spaces v_1 and v_8 are temporarily also considered as design parameter constants.

To begin, we assume that the resulting variety is defined by linear constraint equations, and hence a general linear ansatz polynomial can be written, using

the graded reverse lexicographic monomial ordering [27], as

$$C_1y_3 + C_2y_2 + C_3y_1 + C_4y_0 + C_5x_3 + C_6x_2 + C_7x_1 + C_8x_0 = 0. \tag{13}$$

This linear ansatz polynomial has eight unknown coefficients C_i, $i \in \{1, \cdots, 8\}$. In the case of the left hand side of the RSSR chain, Eq. (11) is substituted into Eq. (13) and after reorganising such that the variable angle parameters of the spherical displacement are collected, yields

$$\begin{aligned}
&(-2C_1d_1v_1 + 2C_3a_1v_1 - 2C_3a_4v_1 + 4C_8v_1 - 2C_4d_1 - 2C_2a_1 + 2C_2a_4 - 4C_5)v_2v_3v_4 \\
&+ (2C_2a_1v_1 + 2C_2a_4v_1 - 2C_4d_1v_1 - 4C_5v_1 + 2C_1d_1 + 2C_3a_1 + 2C_3a_4 - 4C_8)v_2v_3 \\
&+ (2C_1a_1v_1 + 2C_1a_4v_1 + 2C_3d_1v_1 + 4C_6v_1 - 2C_2d_1 + 2C_4a_1 + 2C_4a_4 + 4C_7)v_2v_4 \\
&+ (2C_2d_1v_1 + 2C_4a_1v_1 - 2C_4a_4v_1 - 4C_7v_1 - 2C_1a_1 + 2C_1a_4 + 2C_3d_1 + 4C_6)v_2 \\
&+ (2C_2a_1v_1 + 2C_2a_4v_1 - 2C_4d_1v_1 - 4C_5v_1 + 2C_1d_1 + 2C_3a_1 + 2C_3a_4 - 4C_8)v_3v_4 \\
&+ (2C_1d_1v_1 - 2C_3a_1v_1 + 2C_3a_4v_1 - 4C_8v_1 + 2C_2a_1 - 2C_2a_4 + 2C_4d_1 + 4C_5)v_3 \\
&+ (-2C_2d_1v_1 - 2C_4a_1v_1 + 2C_4a_4v_1 + 4C_7v_1 + 2C_1a_1 - 2C_1a_4 - 2C_3d_1 - 4C_6)v_4 \\
&+ (2C_1a_1v_1 + 2C_1a_4v_1 + 2C_3d_1v_1 + 4C_6v_1 - 2C_2d_1 + 2C_4a_1 + 2C_4a_4 + 4C_7) = 0.
\end{aligned} \tag{14}$$

To fulfil this equation, the coefficients of the motion parameters in Eq. (14) must vanish since the v_2, v_3, and v_4 orientation angle parameters are, in general non-zero. In matrix form, this can be expressed as

$$\begin{bmatrix}
2a_1v_1 + 2a_4v_1 & -2d_1 & 2d_1v_1 & 2a_1 + 2a_4 & 0 & 4v_1 & 4 & 0 \\
-2d_1v_1 & -2a_1 + 2a_4 & 2a_1v_1 - 2a_4v_1 & -2d_1 & -4 & 0 & 0 & 4v_1 \\
-2a_1 + 2a_4 & 2d_1v_1 & 2d_1 & 2a_1v_1 - 2a_4v_1 & 0 & 4 & -4v_1 & 0 \\
2a_1v_1 + 2a_4v_1 & -2d_1 & 2d_1v_1 & 2a_1 + 2a_4 & 0 & 4v_1 & 4 & 0 \\
2d_1 & 2a_1v_1 + 2a_4v_1 & 2a_1 + 2a_4 & -2d_1v_1 & -4v_1 & 0 & 0 & -4 \\
2d_1 & 2a_1v_1 + 2a_4v_1 & 2a_1 + 2a_4 & -2d_1v_1 & -4v_1 & 0 & 0 & -4 \\
2d_1v_1 & 2a_1 - 2a_4 & -2a_1v_1 + 2a_4v_1 & 2d_1 & 4 & 0 & 0 & -4v_1 \\
2a_1 - 2a_4 & -2d_1v_1 & -2d_1 & -2a_1v_1 + 2a_4v_1 & 0 & -4 & 4v_1 & 0
\end{bmatrix}
\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{bmatrix} =
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Solving for the unknown C_i and back-substituting their solutions into the general linear ansatz polynomial Eq. (13) reveals all four hyperplanes that satisfy the variety in $\mathbb{P}^7$. The solution shows that C_1, C_3, C_4, and C_8 are all free parameters with arbitrary values while C_2, C_5, C_6, and C_7 are expressions containing only v_1 and the design parameters and, after simplifying, are each linear in four of the Study parameters, and therefore hyperplanes. These four hyperplanes collected in terms of the Study parameters are

$$\begin{aligned}
0 &= (a_1^2v_1^2 - a_4^2v_1^2 + d_1^2v_1^2 + a_1^2 - a_4^2 + d_1^2)x_3 + (-2d_1v_1^2 - 2d_1)y_0 \\
&\quad + 4a_1v_1y_1 + (2a_1v_1^2 - 2a_4v_1^2 - 2a_1 - 2a_4)y_2, \qquad (15) \\
0 &= (a_1^2v_1^2 - a_4^2v_1^2 + d_1^2v_1^2 + a_1^2 - a_4^2 + d_1^2)x_2 - 4a_1v_1y_0 + (-2d_1v_1^2 - 2d_1)y_1 \\
&\quad + (-2a_1v_1^2 + 2a_4v_1^2 + 2a_1 + 2a_4)y_3, \qquad (16) \\
0 &= (a_1^2v_1^2 - a_4^2v_1^2 + d_1^2v_1^2 + a_1^2 - a_4^2 + d_1^2)x_1 + (2a_1v_1^2 + 2a_4v_1^2 - 2a_1 + 2a_4)y_0 \\
&\quad + (2d_1v_1^2 + 2d_1)y_2 - 4a_1v_1y_3, \qquad (17)
\end{aligned}$$

$$0 = (a_1^2v_1^2 - a_4^2v_1^2 + d_1^2v_1^2 + a_1^2 - a_4^2 + d_1^2)x_0 + (-2a_1v_1^2 - 2a_4v_1^2 + 2a_1 - 2a_4)y_1 + 4a_1v_1y_2 + (2d_1v_1^2 + 2d_1)y_3. \tag{18}$$

The same procedure can be done with the right hand side of the RSSR by substituting Eq. (12) in the general linear ansatz polynomial, Eq. (13). In this case, the motion parameters to be eliminated are v_5, v_6 and v_7. Solving the resulting homogeneous matrix equation for the new unknown C_i yields the following four hyperplanes in a similar way. They are

$$\begin{aligned}0 = {} & (a_7^2\alpha_8^2v_8^2 - 2a_7a_8\alpha_8^2v_8^2 + a_8^2\alpha_8^2v_8^2 + \alpha_8^2d_8^2v_8^2 + a_7^2v_8^2 - 2a_8a_7v_8^2 \\ & + a_8^2v_8^2 + d_8^2v_8^2 + \alpha_8^2a_7^2 + 2a_7a_8\alpha_8^2 + a_8^2\alpha_8^2 + \alpha_8^2d_8^2 + a_7^2 + 2a_7a_8 + a_8^2 + d_8^2)x_3 \\ & + (-2\alpha_8^2d_8v_8^2 + 2d_8v_8^2 + 8a_7\alpha_8v_8 - 2\alpha_8^2d_8 + 2d_8)y_0 \\ & + (-4d_8\alpha_8v_8^2 - 4\alpha_8^2a_7v_8 + 4a_7v_8 - 4d_8\alpha_8)y_1 \\ & + (-2a_7\alpha_8^2v_8^2 + 2\alpha_8^2a_8v_8^2 - 2a_7v_8^2 + 2a_8v_8^2 + 2a_7\alpha_8^2 + 2\alpha_8^2a_8 + 2a_7 + 2a_8)y_2,\end{aligned} \tag{19}$$

$$\begin{aligned}0 = {} & (a_7^2\alpha_8^2v_8^2 - 2a_7a_8\alpha_8^2v_8^2 + a_8^2\alpha_8^2v_8^2 + \alpha_8^2d_8^2v_8^2 + a_7^2v_8^2 - 2a_8a_7v_8^2 \\ & + a_8^2v_8^2 + d_8^2v_8^2 + \alpha_8^2a_7^2 + 2a_7a_8\alpha_8^2 + a_8^2\alpha_8^2 + \alpha_8^2d_8^2 + a_7^2 + 2a_7a_8 + a_8^2 + d_8^2)x_2 \\ & + (4d_8\alpha_8v_8^2 + 4\alpha_8^2a_7v_8 - 4a_7v_8 + 4d_8\alpha_8)y_0 \\ & + (-2\alpha_8^2d_8v_8^2 + 2d_8v_8^2 + 8a_7\alpha_8v_8 - 2\alpha_8^2d_8 + 2d_8)y_1 \\ & + (2a_7\alpha_8^2v_8^2 - 2\alpha_8^2a_8v_8^2 + 2a_7v_8^2 - 2a_8v_8^2 - 2a_7\alpha_8^2 - 2\alpha_8^2a_8 - 2a_7 - 2a_8)y_3,\end{aligned} \tag{20}$$

$$\begin{aligned}0 = {} & (a_7^2\alpha_8^2v_8^2 - 2a_7a_8\alpha_8^2v_8^2 + a_8^2\alpha_8^2v_8^2 + \alpha_8^2d_8^2v_8^2 + a_7^2v_8^2 - 2a_8a_7v_8^2 \\ & + a_8^2v_8^2 + d_8^2v_8^2 + \alpha_8^2a_7^2 + 2a_7a_8\alpha_8^2 + a_8^2\alpha_8^2 + \alpha_8^2d_8^2 + a_7^2 + 2a_7a_8 + a_8^2 + d_8^2)x_1 \\ & + (-2a_7\alpha_8^2v_8^2 + 2\alpha_8^2a_8v_8^2 - 2a_7v_8^2 + 2a_8v_8^2 + 2a_7\alpha_8^2 + 2\alpha_8^2a_8 + 2a_7 + 2a_8)y_0 \\ & + (2\alpha_8^2d_8v_8^2 - 2d_8v_8^2 - 8a_7\alpha_8v_8 + 2\alpha_8^2d_8 - 2d_8)y_2 \\ & + (4d_8\alpha_8v_8^2 + 4\alpha_8^2a_7v_8 - 4a_7v_8 + 4d_8\alpha_8)y_3,\end{aligned} \tag{21}$$

$$\begin{aligned}0 = {} & (a_7^2\alpha_8^2v_8^2 - 2a_7a_8\alpha_8^2v_8^2 + a_8^2\alpha_8^2v_8^2 + \alpha_8^2d_8^2v_8^2 + a_7^2v_8^2 - 2a_8a_7v_8^2 \\ & + a_8^2v_8^2 + d_8^2v_8^2 + \alpha_8^2a_7^2 + 2a_7a_8\alpha_8^2 + a_8^2\alpha_8^2 + \alpha_8^2d_8^2 + a_7^2 + 2a_7a_8 + a_8^2 + d_8^2)x_0 \\ & + (2a_7\alpha_8^2v_8^2 - 2\alpha_8^2a_8v_8^2 + 2a_7v_8^2 - 2a_8v_8^2 - 2a_7\alpha_8^2 - 2\alpha_8^2a_8 - 2a_7 - 2a_8)y_1 \\ & + (-4d_8\alpha_8v_8^2 - 4\alpha_8^2a_7v_8 + 4a_7v_8 - 4d_8\alpha_8)y_2 \\ & + (2\alpha_8^2d_8v_8^2 - 2d_8v_8^2 - 8a_7\alpha_8v_8 + 2\alpha_8^2d_8 - 2d_8)y_3.\end{aligned} \tag{22}$$

Solving Eqs. (15), ..., (18) for the four y_i and substituting these expressions into Eqs. (19), ..., (22) leaves four equations in the four unknown Study parameters x_i. This suggests solving the system of four equations for the four unknown x_i. However, doing so leads only to the trivial solution $x_i = y_i = 0$, $i \in \{0, 1, 2, 3\}$, which we call the null point. This result can be explained geometrically in $\mathbb{P}^7$ as follows: the two special 3-spaces representing the displacements of the S joints are two $SO(3)$ A-planes that are moved around on S_6^2 under the action of the two R joints, and only ever intersect in the null point.

But, there is a solution. Further inspection of the four equations shows that the equations form a homogeneous system of linear equations. Expressing this linear homogeneous system in matrix-vector form $\mathbf{Cx} = \mathbf{0}$, we know that this system only has a nontrivial solution when the determinant of the 4×4 coefficient matrix $\mathbf{C}$ with respect to the x_i vanishes [28]. Thus, after computing the determinant and omitting the factors that can never vanish, the general algebraic IO equation of the RSSR linkage arises directly from the determinant as

$$\begin{aligned} &Av_1^2v_8^2 + 8d_1\alpha_8a_7v_1^2v_8 + 8d_8\alpha_8a_1v_1v_8^2 + Bv_1^2 \\ &+8a_1a_7(\alpha_8 - 1)(\alpha_8 + 1)v_1v_8 + Cv_8^2 + 8d_8\alpha_8a_1v_1 + 8d_1\alpha_8a_7v_8 + D = 0, \end{aligned} \tag{23}$$

where

$$\begin{aligned} A &= (\alpha_8^2 + 1)A_1A_2 + R, \\ B &= (\alpha_8^2 + 1)B_1B_2 + R, \\ C &= (\alpha_8^2 + 1)C_1C_2 + R, \\ D &= (\alpha_8^2 + 1)D_1D_2 + R, \end{aligned}$$

and

$$\begin{aligned} A_1 &= (a_1 - a_4 + a_7 - a_8), & A_2 &= (a_1 + a_4 + a_7 - a_8), \\ B_1 &= (a_1 + a_4 - a_7 - a_8), & B_2 &= (a_1 - a_4 - a_7 - a_8), \\ C_1 &= (a_1 - a_4 - a_7 + a_8), & C_2 &= (a_1 + a_4 - a_7 + a_8), \\ D_1 &= (a_1 + a_4 + a_7 + a_8), & D_2 &= (a_1 - a_4 + a_7 + a_8), \end{aligned}$$

with

$$R = (d_1 - d_8)^2\alpha_8^2 + (d_1 + d_8)^2.$$

Eq. (23) is an implicit biquadratic algebraic curve of degree 4 in the joint angle parameters v_1 and v_8, as one would expect.

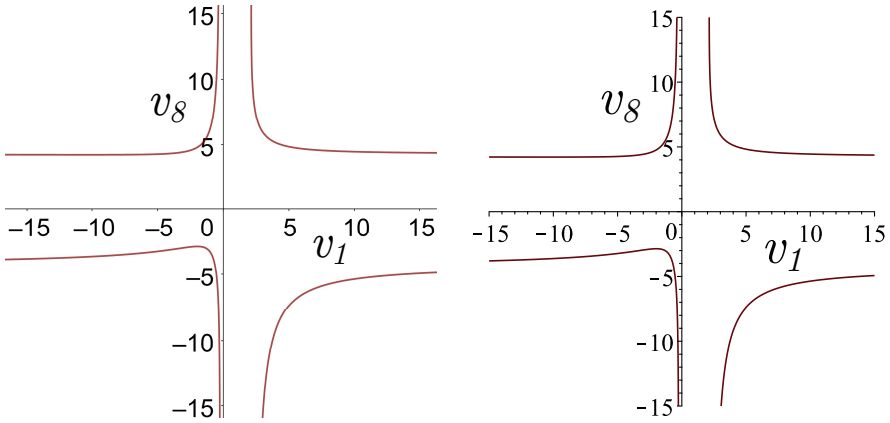
5. Numerical Approach

The degree four algebraic IO equation for the RSSR expressed as Eq. (23) will be compared to the result from a concomitant numerical method. The aim for the numerical method is to compute an eliminant with the general approach of numerical elimination theory [29, Ch. 16]. This involves performing computations using the given polynomial system from Eq. (10) and geometrically projecting points via pseudowitness sets [30]. For this problem, the pseudowitness set provided that the degree of the eliminant is 8 in 9 variables $(v_1, v_8, \alpha_8, a_1, a_4, a_7, a_8, d_1, d_8)$. Since there are a total of $\binom{9+8}{8} = 24310$ monomials of degree at most 8 in 9 variables, the approach is to use the pseudowitness set to generate at least 24310 sample points and then to use interpolation to

recover the eliminant [31, Ch. 6]. To gather these sample points, one randomly fixes values of the parameters $\alpha_8, a_1, a_4, a_7, a_8, d_1, d_8$, and solves for the angle parameter values, v_1 and v_8 using any of a variety of sampling methods within numerical algebraic geometry [32, Sec. 2.3]. This yields precisely the same IO equation as the linear implicitisation approach, Eq. (23).

6. Geometric Verification

To verify both the algebraic and numerical results, the IO equation of an arbitrary linkage was animated in GeoGebra. The model enabled measurement of the output angle for any given input angle. Tracing the locus of each input-output pair results in a curve which is compared with the herein derived IO equation, Eq. (23). The chosen design parameters for the example linkage are $a_1 = 3$, $a_4 = 5$, $a_7 = 9$, $a_8 = 11$, $d_1 = 1$, $d_8 = 3$, and $\tau_8 = 60°$. While the



(a) IO equation generated in GeoGebra. (b) Derived IO equation according to Eq. (23).

Figure 3: Example RSSR function generator with $a_1 = 3$, $a_4 = 5$, $a_7 = 9$, $a_8 = 11$, $d_1 = 1$, $d_8 = 3$, and $\tau_8 = 60°$.

result of the GeoGebra file is displayed in Fig. 3a, substituting the same design parameters into Eq. (23) yields the curve in Fig. 3b. As can be seen, the curves are congruent which further suggests that Eq. (23) is indeed correct.

7. Applications

To demonstrate how the form of the algebraic IO equation for the RSSR linkage that has been obtained by the methods outlined in this paper is particularly useful for mechanical design by way of synthesis and analysis of RSSR mechanisms, several applications will be summarised and illustrated with examples. While it must be acknowledged that the IO equation itself is not new, see the 1955 book by J.S. Beggs for example [1], the algebraic form leads to computationally efficient and mathematically elegant tools for synthesis and analysis

of RSSR linkages that are entirely new, and will be reported for the first time in what follows.

7.1. Continuous Approximate Synthesis of RSSR Function Generators Minimising the Design and Structural Errors

For an RSSR function generator linkage, the synthesis equation will contain six of the seven DH design parameters a_1, a_4, a_7, d_1, and d_8, all normalised by $a_8 = 1$, along with the twist parameter α_8. Exact synthesis results in a linkage that precisely generates the prescribed function, but only for the six precision IO pairs, which are used to generate a set of six synthesis equations linear in the six unknown DH link design parameters. In a highly relevant paper from 1973 [15], the trigonometric form of the RSSR IO equation is derived in the style of Freudenstein for the first time, leading to a synthesis equation with all seven of the DH design parameters which was successfully applied to exact function generator synthesis.

Whereas, approximate synthesis uses $n > 7$ precision IO pairs to create an overconstrained set of synthesis equations leading to a linkage that generates the desired function, in general, but only approximately over the desired displacement range due to errors induced by the number of precision pairs as well as their spacing. *Design* and *structural* errors [14] are important performance indicators used in the assessment and optimisation of mechanical systems intended as function generating linkages designed by means of approximate synthesis. The design error is the residual of the identified linkage in satisfying the synthesis equation [16], and is evaluated at each of $n > 7$ precision points in a discrete set satisfying the prescribed function. Minimising the Euclidean norm of the design error leads to a linear least-squares problem. The structural error, on the other hand, is defined as the difference between the prescribed output angle, and the output angle that is generated by the linkage at each precision point [33]. This problem is typically solved by minimising the norm of the array of the structural error evaluated at each precision point using some form of Gauss-Newton non-linear minimisation, requiring an iterative solution procedure that terminates when a desired minimum norm threshold is obtained. The structural error is arguably the metric that truly matters since it is directly related to the physical performance of the linkage.

However, it was observed in [34] that as the cardinality of the data set used to compute the design error minimising linkage becomes large, on the order of $n \geq 40$, the design error minimising linkage tends to converge to the structural error minimising linkage. Hence, one may avoid the non-linear structural error computation provided a sufficient number of precision points are specified. Continuous approximate synthesis eliminates the problem of determining an appropriate cardinality for the data-set because it evaluates the case for $n \to \infty$. Hence there is no need to search for some convergence in order to set an appropriate value of n, which eliminates a source of error. Unfortunately, while it was demonstrated in [35] that this extension is possible through the integration of the trigonometric Freudenstein equation for planar 4R linkages, the generalisation of the process is computationally prohibitive and any advantage obtained

through the elimination of the need for an explicit solution to the non-linear structural error problem is lost to the numerical complexity of the integration. The planar 4R continuous approximate synthesis example presented in [35] employed the Matlab function *quadl*, which employs recursive adaptive Lobatto quadrature [36], and the computation time to approximately evaluate the integral required more than four hours on an Intel 32-bit dual-core x86 CPU @ 3.10 GHz. The relative complexity involved in integrating the trigonometric RSSR IO equation would likely require an order of magnitude more computation time than that required to integrate that of the 4R IO equation.

While there are many algebraic, meaning non-trigonometric, methods for approximate synthesis in the vast body of literature, see [37] for but one example, there are none which integrate the synthesis equations, thereby making the cardinality of the IO data set tend towards infinity. The following method integrates the square of Eq. (23) between the lower and upper input angular range limits generating a continuous infinite set of IO angle pairs. We partition the result into a 25x1 array of angle parameters which we call the synthesis array $\mathbf{s}$, and a 25x1 array of the seven associated DH link coefficients of $a_1, a_4, a_7, a_8, d_1, d_8$ and the twist parameter α_8, which we call the design parameter array $\mathbf{p}$. Substitute the prescribed function $v_8 = f(v_1)$ into the synthesis array, $\mathbf{s}$. To establish the synthesis equation, which is now a function of only v_1, take the Euclidean inner product of $\mathbf{p}$ with the integral of $\mathbf{s}$ over the prescribed bounds for v_1. The result of this inner product is then minimised over the real numbers. The output of this method is the seven link DH parameters that minimise both the design and structural errors for the RSSR linkage in generating the prescribed $v_8 = f(v_1)$ function, which is summarised by

$$\min_{(a_1,a_4,a_7,a_8,d_1,d_8,\alpha_8)\in\mathbb{R}} \left(\mathbf{p} \cdot \int_{v_{1_{\min}}}^{v_{1_{\max}}} \mathbf{s}(v_1, f(v_1)) dv_i \right) = 0. \tag{24}$$

The *Minimize* command used in Maple 2021 to solve the problem computes a local minimum of an objective function subject to constraints. If the problem is convex, as when the objective function and constraints are linear, for example, the solution will also be a global minimum. The algorithms that this command use assume the objective function and constraints are twice continuously differentiable. In this context, the Euclidean norm of the structural error over every point in the generated function is nothing more than the area between the prescribed function and the generated function in the variable angle parameter plain, which is equivalent to the design error.

An example will now be considered. Let the prescribed function be

$$v_8 = 2 + \tan\left(\frac{v_1}{{v_1}^2 + 1}\right), \quad 0 \leq v_1 \leq 2. \tag{25}$$

The first step is to compute Eq. (24). Then, to obtain a reasonable initial guess for the *Minimize* command used in Maple 2021, six IO pairs $[v_1, v_8]$ were

specified for the exact synthesis problem with $a_8 = 1$ as

$$[0, 2], \left[\frac{1}{5}, \frac{6560}{2989}\right], \left[\frac{1}{4}, \frac{30055}{13419}\right], \left[\frac{1}{3}, \frac{31661}{3710}\right], \left[\frac{1}{2}, \frac{49597}{0471}\right], \left[1, \frac{64699}{25409}\right].$$

The exact synthesis results are listed in Table 2. Using those design parameters as initial guesses for the *Minimize* command lead to the parameter values also listed in Table 2. The run time needed by Maple 2021 to integrate Eq. (24) on a 64-bit Intel Core i7-7700 CPU @ 3.60 GHz was 9.23 seconds, while the time required to run the *Minimize* command was less than 0.01 seconds. This runtime is impressive given that the computations were preformed using symbolic computer algebra compared to the 4 hours to run the numerical integration for a planar 4R linkage using Matlab; granted that this computation was performed on a much older Intel 32-bit dual-core x86 CPU @ 3.10 GHz. Regardless, the trigonometric integration required three orders of magnitude more computation time for the planar 4R synthesis compared to time required for the spatial RSSR synthesis.

Table 2: Synthesis results.

DH parameters	Exact synthesis	Continuous approximate synthesis
a_1	-0.5469961643	-0.481883141397214
a_4	3.760575070	3.76405010790231
a_7	1.349675373	1.35343558991690
a_8	1	0.957062422213279
α_8	0.8098696692	0.807467792413472
d_1	-4.899249807	-4.89575807959238
d_8	1.499319150	1.58161616407823

Table 3: Structural error generating Eq. (25).

Structural error	Exact synthesis	Continuous approximate synthesis
	0.011635738	-0.000261858

A comparison of the prescribed function Eq. (25) with the exact and continuous approximate synthesis generated functions in the v_1-v_8 plane are shown in Fig. 4. The structural and design errors are simply the difference of the areas under the IO curves. It is to be seen that the structural error for the continuous approximate synthesis linkage is two orders of magnitude smaller that the structural error for the linkage generated using exact synthesis, see Table 3.

7.2. Mobility Limits

With the algebraic IO equation, it is a very simple matter to determine general conditions for the relative mobility of the two ground fixed links. Treating the v_1-v_8 IO pair to be coordinate axes in the plane spanned by the two, then

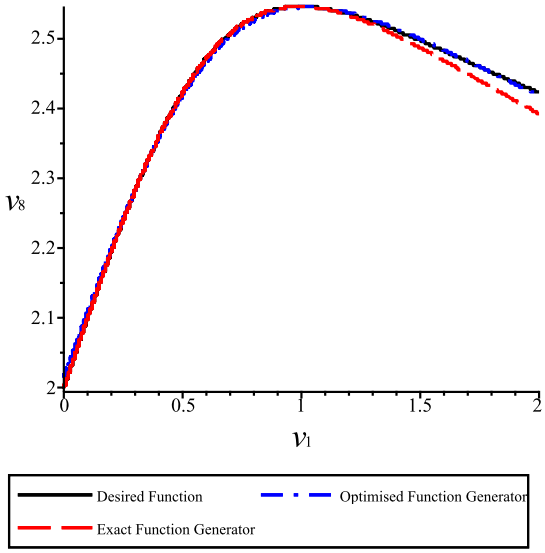


Figure 4: The prescribed, exact, and continuous synthesis approximation of Eq. (25) in the v1-v8 plane.

the IO equation contains two double points at infinity on the v_1 and v_8 axes. The double points at infinity belonging to each of the coordinate axes together with the ability of links a_1 and a_7 to reach $v_1 = 0$ and $v_8 = 0$ completely define the mobility limits, if they exist, between the v_1-v_8 angle parameter pair. The examination of this is sufficient to determine whether a particular joint enables a crank, a rocker, a π-rocker, or a 0-rocker link motion. See [4] from 1971 for the first double point analysis of the RSSR, but only for a simplified special case, where mobility criteria, though incomplete, are reported.

We proceed by evaluating whether each double point has a pair of real, or complex conjugate tangents. If the double point has two real distinct tangents, it is a crunode; if it has two real coincident tangents, it is a cusp; and if the tangents are both complex conjugates, the double point is an acnode. Thus, after homogenising the v_1-v_8 IO equation using the homogenising coordinate v_0, leading to IO_h, the following discriminant yields information on the double point at infinity on the v_i axis:

$$\Delta_{v_i} = \left(\frac{\partial^2 IO_h}{\partial v_j \partial v_0}\right)^2 - \frac{\partial^2 IO_h}{\partial v_j^2}\frac{\partial^2 IO_h}{\partial v_0^2} \begin{cases} > 0 \Rightarrow \text{crunode;} \\ = 0 \Rightarrow \text{cusp;} \\ < 0 \Rightarrow \text{acnode.} \end{cases} \tag{26}$$

The values for Δ_{v_1} and Δ_{v_8} are obtained by substituting values for the seven variable link DH design parameters into the following two discriminants:

$$\Delta_{v_1} = 32a_7^2\alpha_8^2 d_1^2 - AB; \tag{27}$$

and

$$\Delta_{v_8} = 32a_1^2\alpha_8^2 d_8^2 - AC, \tag{28}$$

Table 4: Mobility of a_1 and a_7.

Δ_{v_1}	Ω_{v_1}	mobility of a_1	Δ_{v_8}	Ω_{v_8}	mobility of a_7
≥ 0	≥ 0	crank	≥ 0	≥ 0	crank
≥ 0	< 0	π-rocker	≥ 0	< 0	π-rocker
< 0	≥ 0	0-rocker	< 0	≥ 0	0-rocker
< 0	< 0	rocker	< 0	< 0	rocker

where A, B, and C are all defined in Eq. (23).

From these conditions we can extract information on the ability of the input and output links to rotate through π. For example, if $\Delta_{v_1} \geq 0$, then the double point at $v_1 = \infty$ is either a crunode or a cusp. Knowing that $v_1 = \infty$ corresponds to $\theta_1 = 180°$, this implies that the link a_1 can rotate through π. Similarly, if $\Delta_{v_1} < 0$, then the double point at $v_1 = \infty$ is an acnode which in turn indicates that a_1 has a rotation limit less than π.

It is equally required to investigate whether the linkage is assemblable at $v_i = 0$. Clearly, one possibility to obtain a condition with this information can be derived using the v_1-v_8 equation by substituting $v_i = 0$ and solving for v_j. For each of v_1 and v_8 we obtain a radicand Ω_v whose value must be $\Omega_v \geq 0$ if the link can rotate through 0:

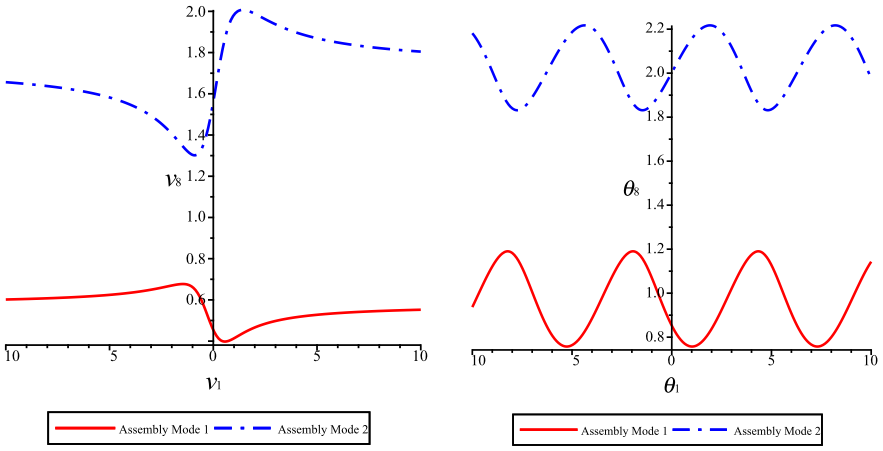
$$\begin{aligned}\Omega_{v_1} &= -\alpha_8^2 (d_1 - d_8)^2 \left(\alpha_8^2 (d_1 - d_8)^2 + 2 (d_1 + d_8)^2\right) - (d_1 + d_8)^4 + \\ &\quad 2\big(a_1^2+2a_1a_8-a_4^2+a_7^2+a_8^2\big)\big[4d_8\big(\alpha_8^4 d_1-\alpha_8^2 d_8-d_1\big)-\big(d_1^2+d_8^2\big)\big(\alpha_8^4+1\big)\big] + \\ &\quad 4\left(a_1^2 + 2a_1a_8 - a_4^2 - 3a_7^2 + a_8^2\right)\alpha_8^2 d_1^2 - C_1C_2D_1D_2\left(\alpha_8^2 + 1\right)^2; \qquad (29)\end{aligned}$$

$$\begin{aligned}\Omega_{v_8} &= -\alpha_8^2 (d_1 - d_8)^2 \left(\alpha_8^2 (d_1 - d_8)^2 + 2 (d_1 + d_8)^2\right) - (d_1 + d_8)^4 + \\ &\quad 2\big(a_1^2-a_4^2+a_7^2+2a_7a_8+a_8^2\big)\big[4d_1\big(d_8\alpha_8^4-\alpha_8^2 d_1-d_8\big)-\big(d_1^2+d_8^2\big)\big(\alpha_8^4+1\big)\big] + \\ &\quad 4\left(3a_1^2 + a_4^2 - a_7^2 - 2a_7a_8 - a_8^2\right)\alpha_8^2 d_8^2 - B_1B_2D_1D_2\left(\alpha_8^2 + 1\right)^2; \qquad (30)\end{aligned}$$

where B_1, B_2, C_1, C_2, D_1, and D_2 are all defined in Eq. (23). With this information we have a completely general classification scheme to determine the relative mobilities of the RSSR, see Table 4.

We will verify the mobility classification with an example. Let the DH parameters be $a_1 = 1/8$, $a_4 = 4$, $a_7 = 1$, $a_8 = 1/8$, $\alpha_8 = \tan((60\pi/180)/2)$, $d_1 = 2$, $d_8 = 2$. Evaluating the discriminants with these DH link design parameters reveals that $\Delta_{v_1} = 10.6667$ and $\Omega_{v_1} = 6.437500000$, indicating that a_1 has no mobility limits while $\Delta_{v_8} = -36$ and $\Omega_{v_8} = -12.6667$, indicating that a_7 is a rocker in each assembly mode. It is a simple matter to determine the configuration and the extreme values of v_8 by evaluating the appropriate derivatives

of Eq. (23) for each assembly mode. The corresponding IO curves in both the v_1-v_8 and θ_1-θ_8 planes are illustrated in Fig. 5 which validates the classification.



(a) IO curve in v_1-v_8 plane.

(b) IO curve in θ_1-θ_8 plane.

Figure 5: RSSR mobility for $a_1 = 1/8$, $a_4 = 4$, $a_7 = 1$, $a_8 = 1/8$, $\alpha_8 = \tan((60\pi/180)/2)$, $d_1 = 2$, $d_8 = 2$.

7.3. Differential Kinematics

Finally, we will determine the extreme output angular velocities and accelerations for a constant input angular velocity. These are important for bearing sizing, among other design considerations that are vital to robust mechanical design of RSSR linkages. While there have been some investigations in the literature examining the dynamics of the RSSR, there are no straightforward methods to be found that give explicit algebraic equations for computing the angular velocity and acceleration extrema for the RSSR, see [6, 38] for example. In [6] the time derivatives of the Cartesian coordinates of the joint centres are used instead of the angles that define the orientation of the links, meaning the equations are not IO equations per se. We therefore believe that the algebraic RSSR IO equation derived in this paper is best suited for such computations. However, to identify extreme angular velocity and acceleration outputs for a constant input angular velocity requires that the angle parameters be transformed back into angles. While $\dot{\theta}_1$ may be constant the corresponding parameter $\dot{v}_1$ is not since it is configuration dependent. That is

$$\dot{v}_1 = \frac{d\tan(\theta_1/2)}{dt} = \frac{\dot{\theta}_1\left(\tan^2(\theta_1/2)+1\right)}{2} = \frac{\dot{\theta}_1(v_1^2+1)}{2}. \quad (31)$$

The first step is to take the first two time derivatives of Eq. (23), which will not be listed here in the interest of brevity. The extreme angular velocities and accelerations, along with the configurations in which they occur in both

assembly modes can be easily obtained computationally with the following two algorithms.

Extreme RSSR angular velocity algorithm.

If values for a_1, a_4, a_7, a_8, d_1, d_8, and α_8 are specified and the input angular velocity is a constant specified value, we wish to determine the critical values $\theta_{1_{\text{crit}}}$ that result in $\dot{\theta}_{8_{\text{min/max}}}$, so we need to eliminate θ_8 from both the position and angular velocity IO equations.

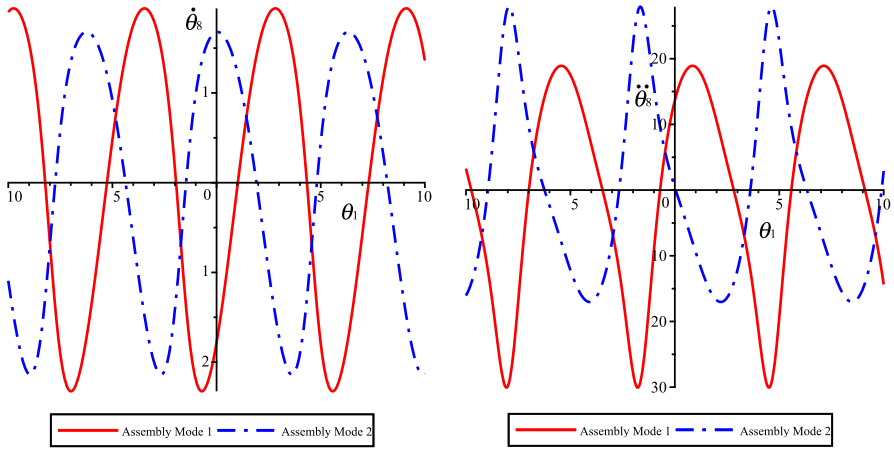
1. Convert v_1 and v_8 in the IO equation to angles as $v_i = \tan(\theta_i/2)$ and solve for θ_8.
2. Substitute the expression for θ_8 from Step 1 into the $\dot{\theta}_1$-$\dot{\theta}_8$ equation and solve for $\dot{\theta}_8$, which gives $\dot{\theta}_8 = f(\theta_1)$ since $\dot{\theta}_1$ is a specified constant.
3. Solve $\dfrac{d\dot{\theta}_8}{d\theta_1} = 0$ for $\theta_{1_{\text{crit}}}$ and determine the values of $\dot{\theta}_{8_{\text{min/max}}}$ corresponding to each distinct value of $\theta_{1_{\text{crit}}}$.

Extreme RSSR angular acceleration algorithm.

If values for a_1, a_4, a_7, a_8, d_1, d_8, and α_8 are specified and the input angular velocity is a constant specified value, we wish to determine the critical values $\theta_{1_{\text{crit}}}$ that result in $\ddot{\theta}_{8_{\text{min/max}}}$, so we need to eliminate θ_8 and $\dot{\theta}_8$ from the position, angular velocity, and acceleration IO equations.

1. Convert v_1 and v_8 in the IO equation to angles as $v_i = \tan(\theta_i/2)$ and solve for θ_8.
2. Substitute the expression for θ_8 from Step 1 into the $\dot{\theta}_1$-$\dot{\theta}_8$ equation and solve for $\dot{\theta}_8$, which gives $\dot{\theta}_8 = f(\theta_1)$ since $\dot{\theta}_1$ is a specified constant.
3. Substitute the expressions for θ_8 and $\dot{\theta}_8$ into the $\ddot{\theta}_1$-$\ddot{\theta}_8$ equation.
4. Solve the resulting equation for $\ddot{\theta}_8$, which gives $\ddot{\theta}_8 = f(\theta_1)$ since $\ddot{\theta}_1 = 0$.
5. Solve $\dfrac{d\ddot{\theta}_8}{d\theta_1} = 0$ for $\theta_{1_{\text{crit}}}$ and determine the values of $\ddot{\theta}_{8_{\text{min/max}}}$ corresponding to each distinct value of $\theta_{1_{\text{crit}}}$.

For an example, again let the DH parameters be $a_1 = 1/8$, $a_4 = 4$, $a_7 = 1$, $a_8 = 1/8$, $\alpha_8 = \tan((60\pi/180)/2)$, $d_1 = 2$, $d_8 = 2$ and the constant input angular velocity be $\dot{\theta}_1 = 10$ rad/s. Using the two algorithms above the output angular velocity and acceleration are expressed in terms of the input angle, see Figs. 6. To the best of the authors collective knowledge, the extreme angular accelerations for an RSSR linkage have not been reported in the literature until now. Even if this is not precisely so, it is clear that the algebraic form of the RSSR equation in the form presented herein has distinct advantages for computation compared to any other representation. The extreme angular accelerations $\ddot{\theta}_{8_{\text{min/max}}}$ and critical input angles are computed and listed in Table 5.



(a) RSSR angular velocity profile.

(b) RSSR angular acceleration profile.

Figure 6: RSSR angular velocity and acceleration profiles for $a_1 = 1/8$,$a_4 = 4$, $a_7 = 1$, $a_8 = 1/8$, $\alpha_8 = \tan((60\pi/180)/2)$, $d_1 = 2$, $d_8 = 2$, $\dot{\theta}_1 = 10$ rad/s.

Table 5: $\ddot{\theta}_{8_{\min/\max}}$ and $\theta_{1_{\text{crit}}}$ for $\dot{\theta}_1 = 10$ rad/s.

Assembly Mode	$\ddot{\theta}_{8_{\min/\max}}$ rad/s^2	$\theta_{1_{\text{crit}}}$ rad
1	-30.06554948	4.506090280
	18.91834314	0.8463167974
2	-17.03055542	2.201742476
	27.91274981	4.631288097

8. Conclusions

Recently it has been shown that Study's kinematic image space and elimination theory provide an excellent, straight forward tool to derive algebraic IO equations for planar, spherical, and Bennett linkages. In this paper, the same method was extended to arbitrary spatial RSSR four-bar linkages. After describing the linkage with standard DH parameters and mapping the closure equation into Study's kinematic image space, the intermediate motion parameters were eliminated with two concomitant methods: algebraically using the linear implicitisation algorithm; and numerically using pseudowitness sets to generate points and then interpolation to recover the eliminant. Both methods lead to the same IO equation containing four more complicated coefficients of the input and output angles compared to the planar 4R, but clearly containing the algebraic IO equation of planar 4R linkages as a subset. This IO equation was additionally verified using a geometric animation in GeoGebra. Finally, three applications were discussed and illustrated with examples to underscore

the utility of the RSSR algebraic IO equation as derived in this paper for four-bar synthesis and analysis.

Acknowledgements

The authors thank Jonathan Hauenstein for helpful discussions regarding numerical computation methods. Martin Pfurner is also acknowledged for thoughtful contributions to our understanding of S_6^2. The authors also acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) via Grant No. RGPIN-2017-06327.

[1] J. Beggs, Mechanism, McGraw-Hill, New York, NY, U.S.A., 1955.

[2] H. Nolle, Ranges of Motion Transfer by the R-G-G-R Linkage, Journal of Mechanisms 4 (1) (1969) 145–157.

[3] F. Sticher, Mobility Limit Analysis of R-S-S-R Mechanisms by "Ellipse Diagram", Journal of Mechanisms 5 (1) (1970) 393–414.

[4] O. Bottema, The Motion of the Skew Four-bar, Journal of Mechanisms 6 (1) (1971) 69–79.

[5] F. Freudenstein, E. Primrose, On the Criteria for the Rotability of the Cranks of a Skew Four-bar Linkage, ASME Journal of Engineering for Industry 98 (4) (1976) 1285–1288.

[6] S. Molian, Kinematics and Dynamics of the RSSR Mechanism, Journal of Mechanisms 6 (1) (1971) 69–79.

[7] J. Duffy, M. Gilmartin, Displacement Analysis of the Generalised RSSR Mechanism, Mechanism and Machine Theory 13 (5) (1978) 533–541.

[8] A. Wilhelm, R. Sodhi, Design of RSSR Function Generator by Curve Matching, in: Unknown Host Publication Title, Japan Soc of Precision Engineers, 1984, pp. 118–123.

[9] C. Mazzotti, M. Troncossi, V. Parenti-Castelli, Dimensional Synthesis of the Optimal RSSR Mechanism for a Set of Variable Design Parameters, Meccanica 52 (1) (2017) 2439–2447.

[10] E. Study, Geometrie der Dynamen, Teubner Verlag, Leipzig, Germany, 1903.

[11] B. Ravani, B. Roth, Motion Synthesis Using Kinematic Mappings, ASME J. Mech., Trans., and Automation 105 (3) (1983) 460–467.

[12] D. R. Walter, M. L. Husty, On Implicitization of Kinematic Constraint Equations, Machine Design & Research (CCMMS 2010) 26 (2010) 218–226.

[13] A. J. Sommese, C. W. Wampler, The Numerical Solution Of Systems Of Polynomials Arising In Engineering And Science, World Scientific Publishing Co. Pte. Ldt., Hackensack, NJ, U.S.A., 2005.

[14] R. S. Hartenberg, J. Denavit, Kinematic Synthesis of Linkages, McGraw-Hill, New York, NY, U.S.A, 1964.

[15] A. V. M. Rao, G. N. Sandor, D. Kohli, A. H. Soni, Closed Form Synthesis of Spatial Function Generating Mechanism for the Maximum Number of Precision Points, ASME J. Eng. Ind. 95 (3) (1973) 725–736.

[16] F. Freudenstein, Design of Four-link Mechanisms, Ph.D. thesis, Columbia University, New York, NY, U.S.A. (1954).

[17] J. Craig, Introduction to Robotics, Mechanics and Control, 4th Edition, Addison-Wesley Publishing Co., Reading, MA, U.S.A., 2017.

[18] B. Mooring, Z. Roth, M. Driels, Fundamentals of Manipulator Calibration, John Wiley & Sons, Inc., New York, NY, U.S.A., 1991.

[19] C. H. An, C. G. Atkeson, J. M. Hollerbach, Model-based Control of a Robot Manipulator, The MIT Press, Cambridge, MA, U.S.A., 1988.

[20] S. Hayati, Robot Arm Geometric Link Parameter Estimation, Proc. 22nd IEEE Conf. on Decision and Control, San Antonio, TX, U.S.A. (1983) 1477–1483.

[21] J. Denavit, R. S. Hartenberg, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, ASME J. Appl. Mech. 22 (2) (1955) 215–221. doi:10.1115/1.4011045.

[22] J. M. Selig, Geometric Fundamentals of Robotics, 2nd Edition, Springer Science + Business Media Inc, New York, NY, U.S.A., 2005.

[23] J. M. Selig, On the Geometry of the Homogeneous Representation for the Group of Proper Rigid-body Displacements, Rom. J. Techn. Sci. – Appl. Mechanics 58 (1-2) (2013) 153–176.

[24] M. Pfurner, The Family of Two Generator 3-Space Rulings of the Study Quadric, Private Communication (August 25, 2022).

[25] M. L. Husty, H.-P. Schröcker, Kinematics and Algebraic Geometry, in: J. M. McCarthy (Ed.), 21st Century Kinematics, Springer-Verlag, London, 2013, pp. 85–123. doi:10.1007/978-1-4471-4510-3_4.

[26] M. L. Husty, D. R. Walter, Mechanism Constraints and Singularities - the Algebraic Formulation, CISM International Centre for Mechanical Sciences 589, Springer International Publishing, Cham, Switzerland, 2019, pp. 101–180. doi:10.1007/978-3-030-05219-5_4.

[27] D. A. Cox, J. Little, D. O'Shea, Using Algebraic Geometry, 2nd Edition, Springer Science + Business Media Inc., New York, NY, U.S.A., 2004.

[28] A. Dresden, Solid Analytic Geometry and Determinants, John Wiley & Sons, Inc., New York, NY, U.S.A., 1930.

[29] D. J. Bates, A. J. Sommese, J. D. Hauenstein, C. W. Wampler, Numerically Solving Polynomial Systems with Bertini, SIAM, Philadelphia, PA, U.S.A., 2013. doi:10.1137/1.9781611972702.

[30] J. D. Hauenstein, A. J. Sommese, Witness Sets of Projections, Applied Mathematics and Computation 217 (7) (2010) 3349–3354. doi:10.1016/j.amc.2010.08.067.

[31] D. Kincaid, W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, American Mathematical Society, Provedince, RI, U.S.A., 2002.

[32] D. A. Cox, Applications of Polynomial Systems, Vol. 134, CBMS Regional Conference Series in Mathematics, American Mathematical Society, Providence, RI, U.S.A., 2020.

[33] S. Tinubu, K. Gupta, Optimal Synthesis of Function Generators Without the Branch Defect, ASME, J. of Mech., Trans., and Autom. in Design 106 (3) (1984) 348–354.

[34] M. Hayes, K. Parsa, J. Angeles, The Effect of Data-set Cardinality on the Design and Structural Errors of Four-bar Function-generators, in: Proceedings of the Tenth World Congress on the Theory of Machines and Mechanisms, *Oulu, Finland*, 1999, pp. 437–442.

[35] A. Guigue, M. Hayes, Continuous Approximate Synthesis of Planar Function-generators Minimising the Design Error, Mechanism and Machine Theory 101 (2016) 158–167.

[36] L. F. Shampine, "Vectorized Adaptive Quadrature in MATLAB", Journal of Computational and Applied Mathematics 211 (February 2008) 131–140.

[37] A. Perez, J. M. McCarthy, Dual Quaternion Synthesis of Constrained Robotic Systems, ASME Journal of Mechanical Design 125 (5) (2004) 425–435.

[38] G. S. Sutherland, Quality of Motion and Force Transmission, Mechanism and Machine Theory 16 (1981) 221–225.