Displacement Analysis of Spherical Mechanisms Having Three or Fewer Loops

Charles W. Wampler

General Motors R&D Center,
Mail Code 480-106-359,
30500 Mound Road,
Warren, MI 48090-9055
e-mail: charles.w.wampler@gm.com

Spherical linkages, having rotational joints whose axes coincide in a common center point, are sometimes used in multi-degree-of-freedom robot manipulators and in one-degree-of-freedom mechanisms. The forward kinematics of parallel-link robots, the inverse kinematics of serial-link robots and the input/output motion of single-degree-of-freedom mechanisms are all problems in displacement analysis. In this article, loop equations are formulated and solved for the displacement analysis of all spherical mechanisms up to three loops. We show how to solve each mechanism type using either a formulation in terms of rotation matrices or quaternions. In either formulation, the solution method is a modification of Sylvester's elimination method, leading directly to numerical calculation via standard eigenvalue routines. [DOI: 10.1115/1.1637653]

1 Introduction

We consider in this paper mechanisms consisting of rigid links with rotational joints between them. Such mechanisms can be classified into three broad categories: planar, having rotational joints with axes all parallel; spherical, having rotational joints whose axes coincide in a common center point; and spatial, being all other cases. Methods for displacement analysis of all *planar* mechanisms having rotational joints have been recently published in [1–3]. This paper provides solution methods for the displacement analysis of all *spherical* mechanisms that have three or fewer loops. These solutions are useful for the forward kinematics of spherical parallel-link robots, the inverse kinematics of spherical serial-link robots (e.g., certain robot wrists) and the input/output motion of spherical single-degree-of-freedom mechanisms. They are also useful in analyzing certain spatial mechanisms whose displacement can be solved by first solving the rotational component of the loop closure equations, followed by a solution for the translational component of the loop equations. One example is the RCCC spatial four-bar; the "Group 1 spatial mechanisms" listed in [4, p. 141] are also examples of this type.

A displacement analysis problem for an *n*-degree-of-freedom mechanism is to find, given the values of *n* input joints, the corresponding values of the remaining output joints. For the purpose of displacement analysis, the two links adjacent to a given input angle can be temporarily replaced by a single rigid link of commensurate dimensions. Thus, displacement analysis reduces to the problem of determining the assembly configurations of a related immobile structure.

The new contribution of this paper is to give simple solutions to the three indecomposable three-loop spherical structures; all other three-loop spherical structures are composites of one-loop and two-loop structures. There is only one type of one-loop spherical structure, the spherical triangle. Its solution is given by the cosine law for spherical triangles, discovered by Johannes Müller (a.k.a., Regiomontanus) in the mid-fifteenth century [5, pp. 237–240]. Solutions for the spherical pentad, the only indecomposable two-loop spherical structure, did not appear until a few years ago [6–8]. Solutions for the input/output relations of certain eight-link spherical mechanisms that reduce to three-loop structures are given in [9]. We include here the solutions to the one and two-loop cases for completeness and to illustrate the general approach before proceeding to the more difficult three-loop cases. Thus,

taken together, the solutions of this paper can be used to solve all three-loop spherical structures and many composite structures having more than three loops. In this paper, all five indecomposable structures up to three loops are formulated and solved in a consistent manner that is easy to implement in a computer program.

Another contribution of the paper is a new elimination procedure, closely related to Sylvester Dialytic Elimination. This procedure facilitates the solution of the problems in hand without introducing extraneous roots.

The paper is organized as follows. First Section 2 reviews the classification of spherical mechanisms into distinct topological types. Then, in Section 3, we discuss the formulation of loop closure equations, both in terms of rotation matrices and quaternions. The new elimination technique is presented in Section 4. We then proceed to solve each mechanism type in Sections 5–7. We present solutions by both rotation matrices and quaternions because each formulation has certain advantages. Each solution is written in terms of a procedure that leads to a generalized eigenvalue problem. In an appendix, a numerical example is given for each problem type.

2 Spherical Structures

As argued in the Introduction, once we have given the values of *n* joints of an *n*-degree-of-freedom mechanism, the determination of the locations of all the links amounts to solving the loop equations of a zero-degree-of-freedom structure. Wherever the value of the joint angle between two links is given, we simply regard the joint as frozen; that is, we replace the pair of links with a single link of commensurate dimensions. In this way, the treatment of all spherical mechanisms immediately reduces to the treatment of just the structures.

Planar mechanisms having only rotational joints can be regarded as special cases of spherical mechanisms in which the common center point is at infinity. Because of this relationship, one finds that a topological classification, which depends only on the number of links and the list of joint connections between them, is identical for planar and spherical linkages. In the correspondence between equivalent planar and spherical linkages, the distance between two rotational joints in a planar link is replaced by the angle between the corresponding joints in the spherical linkage. These angles are the so-called twist angles if one adopts the standard Denavit-Hartenburg description of the linkage. (Link offsets and link lengths are all zero since the joint axes all intersect in a common point.) It is natural to visualize the spherical linkage as a figure drawn on the surface of a sphere, with joint

Fig. 1 Spherical triangle

Fig. 3 Three-loop structure, type 3a
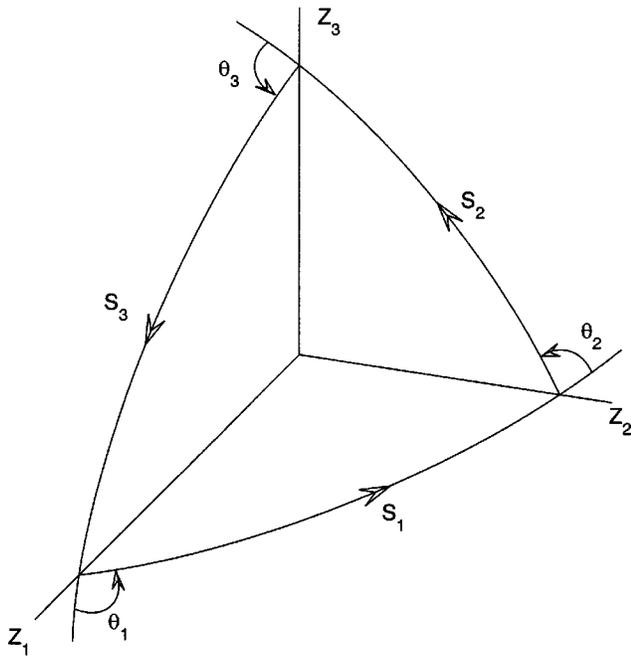
axes as rays from the center that pierce the sphere and with twist angles drawn as great arcs on the sphere. If these arcs are small, the linkage occupies only a tiny portion of the sphere's surface, leading to a planar linkage in the limit as the arc lengths go to zero. This limiting process was used in [10] as a means of synthesizing a planar mechanism by first designing a nearby spherical mechanism.

Structures can be classified as either *decomposable* or *indecomposable*. A decomposable structure contains some proper subset of links that form a structure by themselves; the displacement analysis of such structures can be carried out by first finding all assembly configurations of the substructure and then substituting each of these as a single rigid link into the full structure. After this substitution, the full structure may again be decomposable, but now with fewer links. By such means, the displacement analysis for all possible structures reduces to the displacement analysis of the indecomposable structures. These special structures are at the root of (or equivalent to) a variety of concepts in the kinematics literature, such as "basic kinematic chains," "Assur groups" and "Barranov trusses."

Due to the equivalence of the topological characteristics of planar and spherical structures, the catalog of indecomposable planar structures suffices for the spherical case as well. Complete classifications of three loop structures and how they are built up from the indecomposable structures are reported in [11]. For one loop, we have only one structure: the triangle, shown in Fig. 1. The two-loop and three-loop indecomposable structures are illustrated in Figs. 2–5. In accordance with the number of links in each of these structures, some authors call the triangle a "triad," the two-loop structure the "pentad," and the three-loop structures as "septads" [12]. Illustrations of all the planar structures up to four-loops can be found in [13].

Fig. 2 Two-loop spherical pentad structure

Fig. 4 Three-loop structure, type 3b

$$\hat{Z}_i = \begin{pmatrix} 1-t_i^2 & -2t_i & 0 \\ 2t_i & 1-t_i^2 & 0 \\ 0 & 0 & 1+t_i^2 \end{pmatrix}. \tag{4}$$

With these expressions, we may rewrite Eq. (3) as

$$\mathbf{z}^T S_1 \hat{Z}_2 \cdots \hat{Z}_{k-1} S_{k-1} \mathbf{z} = \mathbf{z}^T S_k^T \mathbf{z}(1+t_2^2) \cdots (1+t_{k-1}^2), \tag{5}$$

which is quadratic in each of the variables $t_2, \ldots, t_{k-1}$.

3.2 Quaternion Formulation. Quaternions can be represented as a sum of scalar coefficients times the four basis elements 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$. It is useful to associate the elements $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ with the x, y, z-axes, respectively, and 1 is just the usual unit scalar. For quaternions, loop equations are still in the form of Eq. (1), but now the quantities in the expression are quaternions, multiplication follows the rules for quaternions, and $I \equiv 1$. Using quaternion notation, a rotation of θ_i about the z-axis is written

$$Z_i = \cos(\theta_i/2) + \mathbf{k}\sin(\theta_i/2) = c_i + \mathbf{k}s_i. \tag{6}$$

Pre-multiplying both sides of Eq. (1) by the conjugate of Z_1, written $Z_1' = c_i - \mathbf{k}s_i$, one obtains

$$S_1 Z_2 S_2 \cdots Z_k S_k = Z_1'. \tag{7}$$

For an arbitrary quaternion $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, define functions that extract individual components as $[\mathbf{i},q] = b$ and $[\mathbf{j},q] = c$. Then, we may write two scalar equations from Eq. (7) as

$$[\mathbf{i}, S_1 Z_2 S_2 \cdots Z_k S_k] = 0,$$
$$[\mathbf{j}, S_1 Z_2 S_2 \cdots Z_k S_k] = 0. \tag{8}$$

In contrast to the rotation matrix formulation, this maneuver has eliminated only one joint rotation, Z_1, but this is compensated by the fact that we obtain two scalar equations per loop. For conciseness below, we will write the two equations for each loop with the abbreviated notation

$$[\{\mathbf{i},\mathbf{j}\}, S_1 Z_2 S_2 \cdots Z_k S_k] = 0. \tag{9}$$

Notice that the equations are homogeneous and linear in the variables $\{c_2, s_2\}, \ldots, \{c_k, s_k\}$.

For exposition below, it is somewhat more convenient to dehomogenize Eq. (9) by dividing through by $c_2 \cdots c_k$ to get

$$[\{\mathbf{i},\mathbf{j}\}, S_1 \hat{Z}_2 S_2 \cdots \hat{Z}_k S_k] = 0, \tag{10}$$

where

$$\hat{Z}_i = Z_i/c_i = 1 + \mathbf{k}t_i. \tag{11}$$

Equation (10) is linear in each of $t_2, \ldots, t_k$.

4 Elimination Procedure

In this paper, we will solve systems of loop equations by a new method that is related to Sylvester's Dialytic Elimination procedure. Sylvester's approach has been used successfully to solve a variety of problems in kinematics, so we begin with a brief synopsis of it. More details and a review of its application to kinematics can be found in [14]. One starts by suppressing one variable, that is, given a system of n polynomials in variables $\{x_1, \ldots, x_n\}$, recast it as polynomials in $\{x_2, \ldots, x_n\}$ with coefficients that depend on x_1. (Since we may renumber the variables at will, there is no loss of generality in assuming x_1 is the hidden variable.) Next, form new polynomials as necessary, by algebraically combining the original polynomials, until one obtains a system with as many equations as monomials. Letting $\mathbf{m}$ be a column vector of the monomials, write the augmented system of equations in matrix form as

$$[A(x_1)]\mathbf{m} = 0. \tag{12}$$

For this system to have a nontrivial solution, ($\mathbf{m} \neq 0$), it is necessary that $\det A(x_1) = 0$. If this determinantal equation is not identi-

Fig. 5 Three-loop structure, type 3c

3 Loop Equations

The first step in modeling a spherical structure is to define a set of coordinate axes at each vertex of each link. We may choose to align the z-axis of each system with the axis of rotation at the joint. The x and y axes can be given any orientation about z, so as to form a right-handed orthogonal coordinate system. Once these arbitrary definitions of the coordinate systems are set, the assumption that the links are rigid implies that the rotation between any two coordinate systems on the same link is constant. We will call such rotations the "side rotations" of the link.

In traversing a loop of a spherical structure, two types of rotations are encountered in alternation: joint rotations, which are variable, and side rotations, which are constant. Hence, a typical loop equation has the form

$$Z_1 S_1 Z_2 S_2 \cdots Z_{k-1} S_{k-1} Z_k S_k = I, \tag{1}$$

where Z_i is a joint rotation, S_i is a side rotation, and I is the identity rotation. One may regard the rotations as either rotation matrices or as quaternions: In either case, the composition of two rotations is found by multiplication. In the following subsections, we will consider both formulations.

3.1 Rotation Matrix Formulation. In this case, in Eq. (1), Z_i and S_i are 3×3 rotation matrices, I is the 3×3 identity matrix, and sequential rotations are compounded by matrix multiplication. Since Z_i is a rotation about the z-axis, it may be written in terms of a rotation angle θ_i as

$$Z_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

Letting $\mathbf{z} = (0\ 0\ 1)^T$, we observe that $\mathbf{z}^T Z_i = \mathbf{z}^T$ and $Z_i \mathbf{z} = \mathbf{z}$. Thus, we may extract from the matrix loop equation, Eq. (1), a single scalar equation that is free of both θ_1 and θ_k as follows:

$$\mathbf{z}^T S_1 Z_2 \cdots Z_{k-1} S_{k-1} \mathbf{z} = \mathbf{z}^T S_k^T \mathbf{z}, \tag{3}$$

where the right-hand side is the (3,3)-element of S_k. Notice that two joint rotations, Z_1 and Z_k, have been eliminated, and we get one scalar equation for each loop.

In what follows, it is convenient to express both $\cos\theta_i$ and $\sin\theta_i$ using the tangent-half-angle formulas $\cos\theta_i = (1-t_i^2)/(1+t_i^2)$ and $\sin\theta_i = 2t_i/(1+t_i^2)$. With these substitutions in mind, define $\hat{Z}_i$ as

cally zero, that is, if the equations in the augmented system are linearly independent, then the solutions of $\det A(x_1)=0$ contain the solutions to the original system. In general, they may contain some extraneous solutions as well, a situation that one wishes to avoid if possible.

In a typical solution in the kinematics literature, the suppression of a variable is done very early, but it is often beneficial to postpone this until after the equations have first been manipulated into a more amenable form. In particular, in [15], several difficult kinematics problems are solved by first computing a graded-degree Gröbner basis and then suppressing a variable as the final step. The variable is chosen so that after suppression the number of equations and monomials are immediately equal. See [16] for background on Gröbner bases.

A similar, but more rigorous and algorithmic, approach has been put forward in [17,18]. After computing a reduced Gröbner basis, one finds the *normal set*, which is the set of monomials that are not divisible by any leading monomial of the basis. One may then derive an eigenvalue problem in which one variable is the eigenvalue and the normal set forms the eigenvector. This is of the same form as Eq. (12), with $A(x_1)$ having a linear dependence on x_1. A key difference, however, is that the eigenvariable x_1 generally appears in the monomials in the eigenvector, that is, x_1 is not completely suppressed.

The new approach described below accomplishes the same outcome as these methods without formally computing a Gröbner basis. We avoid the algorithmic machinery needed to compute a Gröbner basis and instead, for each of the problems at hand, give a simple recipe for generating an augmented system of polynomial equations from the original loop equations. We then identify a set of monomials that play the same role as the normal set in the algorithms mentioned above. The method is no longer general, as the recipes are specific to the problems at hand, but the resulting algorithms are quite transparent and simple to program.

The approach proceeds as follows. First, generate an augmented system of equations by algebraic operations on the original polynomials. In the recipes below, these will be obtained simply by multiplying the polynomials by various monomials. Let $\mathbf{m}$ be the set of all monomials in this augmented system so that it may be expressed in matrix form as $K\mathbf{m}=0$, where K is an $n\times m$, $n<m$, matrix of constant coefficients. It is assumed that we have eliminated any linear dependencies, so that K has full rank n. We choose a variable, say x_1, and a set of $m-n$ monomials $\mathbf{m}_1\subset\mathbf{m}$ and define $\mathbf{m}_2=x_1\otimes\mathbf{m}_1$. These must be chosen such that

1. $\mathbf{m}_2\subset\mathbf{m}$, that is, all x_1 multiples of $\mathbf{m}_1$ are in $\mathbf{m}$,
2. the $m\times m$ system of equations

$$[\hat{K}(x_1)]\mathbf{m}\doteq\begin{pmatrix}K\mathbf{m}\\x_1\mathbf{m}_1-\mathbf{m}_2\end{pmatrix}=0 \qquad (13)$$

is nonsingular for general values of x_1.

Note that $\hat{K}$ is linear in x_1 and its first n rows are constant. This means that $\det \hat{K}(x_1)=0$ is a degree $m-n$ polynomial equation for x_1.

The preferred numerical method for solving Eq. (13) is to extract and solve an order $m-n$ eigenvalue problem whose eigenvector is $\mathbf{m}_1$. To do so, one may partition the monomials $\mathbf{m}$ into four sets: $\mathbf{y}_3=\mathbf{m}_2\setminus\mathbf{m}_1$, $\mathbf{y}_2=\mathbf{y}_3/\mathbf{x}_1$, $\mathbf{y}_1=\mathbf{m}_1\setminus\mathbf{y}_2$, and $\mathbf{y}_4=\mathbf{m}\setminus(\mathbf{m}_1\cup\mathbf{m}_2)$. Partitioning $\hat{K}$ to match, we rewrite Eq. (13) in block matrix form as

$$\begin{pmatrix}K_1 & K_2 & K_3 & K_4\\I_1x+C_1 & C_2 & 0 & 0\\0 & I_2x & -I_2 & 0\end{pmatrix}\begin{pmatrix}\mathbf{y}_1\\\mathbf{y}_2\\\mathbf{y}_3\\\mathbf{y}_4\end{pmatrix}=0, \qquad (14)$$

where I_1,I_2 are identity matrices and C_1,C_2 are sparse, having a single entry of -1 in each row of the pair. In some cases, the last blockwise column is not present, but if it is, it must be full column

rank. Using sparse Gaussian elimination or QR decomposition, K_4 can be reduced to upper triangular form yielding

$$\begin{pmatrix}\tilde{K}_{11} & \tilde{K}_{12} & \tilde{K}_{13} & U\\\tilde{K}_{21} & \tilde{K}_{22} & \tilde{K}_{23} & 0\\I_1x+C_1 & C_2 & 0 & 0\\0 & I_2x & -I_2 & 0\end{pmatrix}\begin{pmatrix}\mathbf{y}_1\\\mathbf{y}_2\\\mathbf{y}_3\\\mathbf{y}_4\end{pmatrix}=0, \qquad (15)$$

for some upper triangular matrix U. Pre-multiplying by the $(m-n)\times m$ matrix

$$\begin{pmatrix}0 & 0 & I_1 & 0\\0 & I_2 & 0 & \tilde{K}_{23}\end{pmatrix},$$

gives the equation

$$\begin{pmatrix}I_1x+C_1 & C_2\\\tilde{K}_{21} & \tilde{K}_{22}+\tilde{K}_{23}x\end{pmatrix}\begin{pmatrix}\mathbf{y}_1\\\mathbf{y}_2\end{pmatrix}=0, \qquad (16)$$

where the trailing blocks have been dropped, since they are zero. The only computation involved is the triangularization of K_4, which can be done efficiently by sparse routines. Eq. (16) is the square generalized eigenvalue problem we seek.

For a particular topological type of spherical mechanism, a specific structure is specified by the values of its "sides," S_i. In each case, one must verify the full-rank condition, number (2) above, for the procedure to be valid. For each type of mechanism, we have verified by numerical test on a structure having generic side rotations that this is so. This is sufficient to demonstrate that the procedure is valid for almost all mechanisms of the type, where the exceptions are an algebraic subset of the whole family. Such exceptional cases require new procedures, often more simple than the general case. We do not explore any such exceptions in this paper.

As in the original Sylvester method, the new approach gives only a necessary condition. It may happen that Eq. (16) allows extraneous roots. However, in the application of the method to spherical mechanisms, we find that in every case we are able to find formulations without any extraneous roots. Again, this has been verified by working numerical examples and checking that all the solutions satisfy the original loop equations.

To solve one-loop and two-loop spherical mechanisms, the original Sylvester method is sufficient, but for the more difficult three-loop mechanisms, the new approach is necessary. For illustrative purposes, we indicate how to solve the simpler cases by the new method before proceeding to the three-loop case.

5 One-Loop Mechanism (Spherical Triangle)

The loop equation for the spherical triangle, Fig. 1, may be written as

$$Z_1S_1Z_2S_2Z_3S_3=I. \qquad (17)$$

Using rotation matrices and Eq. (5), one has

$$\mathbf{z}^TS_1\hat{Z}_2S_2\mathbf{z}=\mathbf{z}^TS_3^T\mathbf{z}(1+t_2^2). \qquad (18)$$

This is a single quadratic equation for t_2. If the sides S_1,S_2,S_3 are taken to be rotations about the respective x-axes, it may be seen that this is equivalent to the classical cosine law for spherical triangles. It should be noted that instead of interior angles, our formulation uses exterior angles at each vertex.

The quaternion formulation Eq. (10) applied to this problem gives

$$[\{\mathbf{i},\mathbf{j}\},S_1\hat{Z}_2S_2\hat{Z}_3S_3]=0. \qquad (19)$$

These are bilinear in t_2 and t_3, that is, only the following monomials appear: $\{1,t_2,t_3,t_2t_3\}$. Following the Modified Sylvester approach from above and choosing $\mathbf{m}_1=\{1,t_2\}$, $\mathbf{m}_2=\{t_3,t_2t_3\}$, one gets a system of the form

$$\begin{pmatrix} A & B \\ t_3 I & -1 \end{pmatrix}\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{pmatrix}=0, \qquad (20)$$

in which A and B are 2×2 matrices of coefficients from the loop equations. Following the formulation of Section 4, $K_2=A$, $K_3=B$, and K_1, K_4 are not present, so we premultiply by $(I\ B)$ to get the 2×2 eigenvalue problem

$$(A+Bt_3)\mathbf{m}_1=0. \qquad (21)$$

In this case, since $\mathbf{m}_2$ is identical to the complement of $\mathbf{m}_1$ (i.e., $\mathbf{m}_2=\mathbf{m}\backslash\mathbf{m}_1$), the modified elimination method amounts to the same thing as the traditional Sylvester Dialytic Method.

6 Two-Loop Mechanism (Spherical Pentad)

The loop equations for the two-loop spherical pentad, Fig. 2, can be written as

$$\begin{aligned} Z_5S_1Z_1S_2Z_2S_3Z_3S_4&=I, \\ Z_6S_5Z_1S_2Z_2S_6Z_4S_7&=I. \end{aligned} \qquad (22)$$

Using rotation matrices, one obtains two quartic equations in t_1,t_2:

$$\begin{aligned} f_1&=\mathbf{z}^T[S_1\hat{Z}_1S_2\hat{Z}_2S_3-S_4^T(1+t_1^2)(1+t_2^2)]\mathbf{z}=0, \\ f_2&=\mathbf{z}^T[S_5\hat{Z}_1S_2\hat{Z}_2S_6-S_7^T(1+t_1^2)(1+t_2^2)]\mathbf{z}=0. \end{aligned} \qquad (23)$$

The total degree is $4^2=16$, but the number of roots is only half that figure, because the equations are bi-quadratic. This can be seen by computing the two-homogeneous Bezout number [19] as the coefficient of $\alpha\beta$ in the polynomial $(2\alpha+2\beta)^2$, which is 8.

To solve this system, first augment it with the equations $f_3=0$, $f_4=0$ where $f_3=t_1f_1$ and $f_4=t_1f_2$. Let $T_0=\{1,t_1,t_1^2,t_1^3\}$. The four polynomials f_1,f_2,f_3,f_4 contain 12 monomials, namely $T_0,T_1=t_2\otimes T_0,T_2=t_2^2\otimes T_0$. In the traditional Sylvester approach, one writes the system in the form $M(t_2)T_0=0$, where M is a 4×4 matrix with entries that are quadratic in t_2. This leads to the eighth-degree polynomial equation $\det M(t_2)=0$.

To apply the method of Section 4, let $\mathbf{m}_1=\{T_0,T_1\}$, and $\mathbf{m}_2=\{T_1,T_2\}$. Append the eight identities $t_2T_0-T_1=0$ and $t_2T_1-T_2=0$, write the equations in the block matrix form as in Eq. (14) and premultiply by a matrix that annihilates T_2 as follows:

$$\begin{pmatrix} 0 & I & 0 \\ I & 0 & K_2 \end{pmatrix}\begin{pmatrix} K_0 & K_1 & K_2 \\ t_2I & -I & 0 \\ 0 & t_2I & -I \end{pmatrix}\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix}=0. \qquad (24)$$

(In this case, K_4 of Eq. (14) does not exist.) Multiply this out and drop away the trailing trivial columns to get an 8×8 eigenvalue problem.

Using quaternions, one obtains two equations per loop, for a total of four equations in t_1,t_2,t_3,t_4:

$$\begin{aligned} f_1&=[\{\mathbf{i},\mathbf{j}\},S_1\hat{Z}_1S_2\hat{Z}_2S_3\hat{Z}_3S_4]=0, \\ f_2&=[\{\mathbf{i},\mathbf{j}\},S_5\hat{Z}_1S_2\hat{Z}_2S_6\hat{Z}_4S_7]=0. \end{aligned} \qquad (25)$$

Note that f_1 does not contain t_4 and f_2 does not contain t_3, and so it is useful to append four new equations:

$$t_4f_1=0, \quad t_3f_2=0. \qquad (26)$$

Altogether, we have 8 equations involving 16 monomials. Let $\mathbf{m}_1=\{1,t_2\}\otimes\{1,t_3\}\otimes\{1,t_4\}$ and $\mathbf{m}_2=t_1\otimes\mathbf{m}_1$, append the 8 identities $t_1\mathbf{m}_1-\mathbf{m}_2=0$ to proceed in a similar fashion as above to get an 8×8 eigenvalue problem.

7 Three-Loop Mechanisms

Three-loop systems are, not surprisingly, a bit more difficult than the one-loop or two-loop systems. In particular, it is no longer simple to find a formulation for which the original

Sylvester Dialytic Method gives a system of minimal degree. Instead, it will lead to a system having extraneous roots, which either cost extra computation to find and eliminate or require additional reduction steps to remove from the analytical equations. In contrast, the new approach of Section 4 leads directly to an eigenvalue problem of minimal size.

7.1 Type 3a. By counterclockwise progression around each of the three loops in Fig. 3, one obtains three-loop equations as follows:

$$\begin{aligned} Z_9S_9Z_2'S_3Z_3S_6Z_6S_{12}&=I, \\ Z_7S_7Z_3'S_1Z_1S_4Z_4S_{10}&=I, \\ Z_8S_8Z_1'S_2Z_2S_5Z_5S_{11}&=I. \end{aligned} \qquad (27)$$

The inverse rotations Z_i' appear where we traverse a joint in one loop in the opposite direction of the adjoining loop. Note that the loop around the central link implies $S_3=(S_1S_2)'$.

The rotational matrix formulation eliminates the six angles $\theta_4,\ldots,\theta_9$, giving the three equations

$$\begin{aligned} f_1&=\mathbf{z}^T[S_9\hat{Z}_2^TS_3\hat{Z}_3S_6-S_{12}^T(1+t_2^2)(1+t_3^2)]\mathbf{z}=0, \\ f_2&=\mathbf{z}^T[S_7\hat{Z}_3^TS_1\hat{Z}_1S_4-S_{10}^T(1+t_1^2)(1+t_3^2)]\mathbf{z}=0, \\ f_3&=\mathbf{z}^T[S_8\hat{Z}_1^TS_2\hat{Z}_2S_5-S_{11}^T(1+t_1^2)(1+t_2^2)]\mathbf{z}=0. \end{aligned} \qquad (28)$$

The three homogeneous Bezout number for this system is the coefficient of $\alpha_1\alpha_2\alpha_3$ in the polynomial $(2\alpha_2+2\alpha_3)(2\alpha_1+2\alpha_3)(2\alpha_1+2\alpha_2)$, which is 16. Note that the equations are numbered such that f_i is missing variable t_i.

An order 16 eigenvalue problem may be derived as follows. Produce an augmented equation set by multiplying each f_i by the 16 monomials $\{1,t_i,t_i^2,t_i^3\}\otimes\{1,t_j\}\otimes\{1,t_k\}$. This gives 48 equations in the 64 monomials $\mathbf{m}=\otimes_{i=1}^3\{1,t_i,t_i^2,t_i^3\}$. Notice that the difference, $64-48=16$, is the size of the eigenvalue problem we seek. We choose t_1 as the eigenvariable and seek a list of 16 monomials $\mathbf{m}_1\subset\mathbf{m}$ such that $t_1\otimes\mathbf{m}_1\subset\mathbf{m}$ and the corresponding matrix $\hat{K}(t_1)$ is generally nonsingular. By numerical test on a generic example, we have verified that

$$\mathbf{m}_1=[\{1,t_1,t_3\}\otimes\{1,t_2,t_2^2,t_2^3\}]\cup\{t_1t_3,t_1t_2t_3,t_3^2,t_2t_3^2\}$$

is sufficient in this regard. Reduction to an order 16 eigenvalue problem proceeds as in Section 4.

We may also solve the problem using quaternions. The quaternion version of the loop equations only eliminates angles $\theta_7,\theta_8,\theta_9$, giving the six equations

$$\begin{aligned} f_1&=[\{\mathbf{i},\mathbf{j}\},S_7\hat{Z}_3'S_1\hat{Z}_1S_4\hat{Z}_4S_{10}]=0, \\ f_2&=[\{\mathbf{i},\mathbf{j}\},S_8\hat{Z}_1'S_2\hat{Z}_2S_5\hat{Z}_5S_{11}]=0, \\ f_3&=[\{\mathbf{i},\mathbf{j}\},S_9\hat{Z}_2'S_3\hat{Z}_3S_6\hat{Z}_6S_{12}]=0. \end{aligned} \qquad (29)$$

At first, it might seem a disadvantage that we have more variables than in the formulation with rotation matrices, but because the equations are lower in degree and more sparse, there is no increase in difficulty. The six-homogeneous Bezout number is still 16. To solve, note that for each f_i, there are three variables among $t_1,\ldots,t_6$ which do not appear: call them t_j,t_k,t_l. Multiply the equations by the eight monomials $\{1,t_j\}\otimes\{1,t_k\}\otimes\{1,t_l\}$ to get $6\cdot8=48$ equations in the 64 monomials $\mathbf{m}=\otimes_{i=1}^6\{1,t_i\}$. Choosing t_1 as the eigenvalue, we find that the monomial set $\mathbf{m}_1=[\{1,t_2,t_3,t_2t_3,t_4,t_2t_4\}\otimes\{1,t_5\}]\cup\{t_6,t_2t_6,t_4t_6,t_2t_4t_6\}$, suffices to derive an order 16 eigenvalue problem.

7.2 Type 3b. The loop equations for Type 3b mechanisms, Fig. 4, are

$$Z_7 S_4 Z_1 S_1 Z_2 S_2 Z_4 S_3 = I,$$

$$Z_8 S_7 Z_3' S_8 Z_2' S_5 Z_5 S_6 = I, \tag{30}$$

$$Z_9 S_{11} Z_1 S_1 Z_2 S_8' Z_3 S_9 Z_6 S_{10} = I.$$

For rotation matrices, these give

$$f_1 = \mathbf{z}^T [S_4 \hat{Z}_1 S_1 \hat{Z}_2 S_2 - S_3^T (1 + t_1^2)(1 + t_2^2)] \mathbf{z} = 0,$$

$$f_2 = \mathbf{z}^T [S_7 \hat{Z}_3^T S_8 \hat{S}_2^T S_5 - S_6^T (1 + t_2^2)(1 + t_3^2)] \mathbf{z} = 0, \tag{31}$$

$$f_3 = \mathbf{z}^T [S_{11} \hat{Z}_1 S_1 \hat{Z}_2 S_8^T \hat{Z}_3 S_9 - S_{10}^T (1 + t_1^2)(1 + t_2^2)(1 + t_3^2)] \mathbf{z} = 0.$$

The appearance of all three variables in f_3 increases the number of roots: the three-homogeneous Bezout number is now 24. The solution procedure is similar to that used for Type 3a mechanisms. First, an augmented system is produced by multiplying f_1, f_2, f_3 by all possible monomials that give degrees less than or equal to 3 for each of the variables. Thus, we get the same 64 monomials as before, but since f_3 now has all three variables at the outset, we get fewer equations. Both f_1 and f_2 give 16 equations in the augmented system, but f_3 gives only 8, for a total of 40 equations. The excess of monomials over equations is $64 - 40 = 24$, so we must append 24 identities to get the desired eigenvalue form of the problem. Choosing to solve for t_3, we find that $\mathbf{m}_1 = [(\{1, t_1, t_1^2, t_1^3\} \otimes \{1, t_2\}) \cup \{t_2^2, t_1 t_2^2, t_2^3, t_1 t_2^3\}] \otimes \{1, t_3\}$, which has 24 elements, suffices to form a nonsingular eigenvalue problem.

For quaternions, the loop equations yield

$$f_1 = [\{\mathbf{i}, \mathbf{j}\}, S_4 \hat{Z}_1 S_1 \hat{Z}_2 S_2 \hat{Z}_4 S_3] = 0,$$

$$f_2 = [\{\mathbf{i}, \mathbf{j}\}, S_7 \hat{Z}_3' S_8 \hat{Z}_2' S_5 \hat{Z}_5 S_6] = 0, \tag{32}$$

$$f_3 = [\{\mathbf{i}, \mathbf{j}\}, S_{11} \hat{Z}_1 S_1 \hat{Z}_2 S_8' \hat{Z}_3 S_9 \hat{Z}_6 S_{10}] = 0.$$

The six-homogeneous Bezout number is 24. To solve, generate an augmented system of equations by multiplying by all the multi-linear monomials in the missing variables. We get the same $2^6 = 64$ monomials as for the Type 3a quaternion formulation, but only $2 \cdot 8 + 2 \cdot 8 + 2 \cdot 4 = 40$ equations. Choosing to solve for t_1, a sufficient choice of monomials is $\mathbf{m}_1 = [(\{1, t_2\} \otimes \{1, t_3\} \otimes \{1, t_4\}) \cup (\{1, t_2, t_2 t_4, t_3 t_4\} \otimes \{t_5\})] \otimes \{1, t_6\}).$

7.3 Type 3c. The loop equations for Type 3c mechanisms, Fig. 5, are

$$Z_7 S_4 Z_1 S_1 Z_2 S_2 Z_4 S_3 = I,$$

$$Z_8 S_8 Z_1 S_1 Z_2 S_5 Z_3 S_6 Z_5 S_7 = I, \tag{33}$$

$$Z_9 S_{11} Z_1 S_1 Z_2 S_5 Z_3 S_9 Z_6 S_{10} = I.$$

For rotation matrices, these give

$$f_1 = \mathbf{z}^T [S_4 \hat{Z}_1 S_1 \hat{Z}_2 S_2 - S_3^T (1 + t_1^2)(1 + t_2^2)] \mathbf{z} = 0,$$

$$f_2 = \mathbf{z}^T [S_8 \hat{Z}_1 S_1 \hat{Z}_2 S_5 \hat{Z}_3 S_6 - S_7^T (1 + t_1^2)(1 + t_2^2)(1 + t_3^2)] \mathbf{z} = 0, \tag{34}$$

$$f_3 = \mathbf{z}^T [S_{11} \hat{Z}_1 S_1 \hat{Z}_2 S_5 \hat{Z}_3 S_9 - S_{10}^T (1 + t_1^2)(1 + t_2^2)(1 + t_3^2)] \mathbf{z} = 0.$$

The three-homogeneous Bezout number now rises to 32. Form an augmented system, as before, with equations having up to the cubic power of each variable, for a total of $16 + 8 + 8 = 32$ equations in 64 monomials. If we choose to solve for t_2, a sufficient choice of monomials is $\mathbf{m}_1 = \{1, t_1, t_1^2, t_1^3\} \otimes \{1, t_2\} \otimes \{1, t_3, t_3^2, t_3^3\}$.

For quaternions, the loop equations yield

$$f_1 = [\{\mathbf{i}, \mathbf{j}\}, S_4 \hat{Z}_1 S_1 \hat{Z}_2 S_2 \hat{Z}_4 S_3] = 0,$$

$$f_2 = [\{\mathbf{i}, \mathbf{j}\}, S_8 \hat{Z}_1, S_1 \hat{Z}_2 S_5 \hat{Z}_3 S_6 \hat{Z}_5 S_7] = 0, \tag{35}$$

$$f_3 = [\{\mathbf{i}, \mathbf{j}\}, S_{11} \hat{Z}_1 S_1 \hat{Z}_2 S_5 \hat{Z}_3 S_9 \hat{Z}_6 S_{10}] = 0$$

The six-homogeneous Bezout number is 32. To solve, generate an augmented system of equations by multiplying by all the multi-linear monomials in the missing variables. We get the same $2^6 = 64$ monomials as before, but only $2 \cdot 8 + 2 \cdot 4 + 2 \cdot 4 = 32$ equations. Choosing to solve for t_1, a sufficient choice of monomials is $\mathbf{m}_1 = \otimes_{i=2}^{6} \{1, t_1\}$.

8 Back-Substitution

Back-substitution is the process of finding all of the joint angles after the basic eigenvalue problem has been solved. In our formulations, the eigenvalues are the solutions for one variable. The eigenvectors are, up to scale, the monomials $\mathbf{m}_1$. In every case, $\mathbf{m}_1$ includes the monomial 1, so the corresponding element in the eigenvector reveals the scale factor. If it is zero, the solution is at infinity; otherwise, we may divide out the scale factor and retrieve the other variables t_i that appear in the equations.

There are, however, some angles that do not appear in any of the equations, because we eliminate them in the initial formulation. For the quaternion formulations, in the notation of Eq. (1), the initial angle θ_1 is eliminated. That angle is easily recovered by inverting the other rotations, that is

$$Z_1 = (S_1 Z_2 S_2 \cdots Z_{k-1} S_{k-1} Z_k S_k)'.$$

For the rotation matrix formulations, two angles θ_1 and θ_k are eliminated from each loop, see Eq. (5), so we must reconstruct both of them. Pre-multiplication of Eq. (1) by $\mathbf{z}^T$ yields 3 equations free of θ_1. For back-substitution, one may use the first two, which are linear in the sine and cosine of θ_k, and therefore give a unique value. (The third of these is exactly Eq. (5).) Once θ_k is known, one may reconstruct the leading rotation in the same way as for the quaternion formulation.

9 Solutions at Infinity

If $\theta = \pi$ then $t = \tan(\theta/2)$ is infinite, so if a mechanism has a solution near π, numerical trouble occurs. There is a simple fix for this problem: replace the angle with $\theta = \phi + c$, for some constant c, and solve for the new variable ϕ. The z-rotation of angle c is constant and can be absorbed into the adjacent side rotations, S. If we choose c at random, there is a zero probability that the solution for ϕ will be exactly π and only a small probability that it will be close enough to π to cause trouble.

A somewhat more elegant approach is to treat the variables homogeneously. For quaternions, this just means that at Eq. (8), we keep $s = \sin(\theta/2)$ and $c = \cos(\theta/2)$ both as variables. In the elimination procedures above, where we multiply by $\{1, t\}$, instead multiply by $\{c, s\}$, and where we use an identity $t\mathbf{m}_1 - \mathbf{m}_2 = 0$, use $s\mathbf{m}_1 - c\mathbf{m}_2 = 0$. The final result will be a generalized eigenvalue problem in c and s, that is, a problem of the form $(Ac + Bs)\mathbf{m} = 0$. When solved by the **qz** algorithm in Matlab, based on [20], this formulation is immune to the $\theta = \pi$ problem. For rotation matrices, a similar treatment may be obtained by usage of the substitutions $\sin \theta = (2sc)/(c^2 + s^2)$ and $\cos \theta = (c^2 - s^2)/(c^2 + s^2)$, which after clearing denominators, leads to homogeneous equations.

It may happen that a problem yields solutions of the form $t = \pm \sqrt{-1}$, or equivalently in the homogeneous treatment $c^2 + s^2 = 0$. These are solutions at infinity of another type: they give no well-defined value for θ. Such solutions indicate that the mechanism under study is a special one with fewer solutions than the general case. This implies that a reduction of the problem should be possible.

10 Conclusion

This paper shows how to formulate and solve the loop equations for all the indecomposable spherical structures up to three loops. Formulations using rotational matrices and quaternions are both considered. In each case, the problem has been transformed into an eigenvalue problem of minimal degree. Numerical examples of each have been solved and found to be well behaved. The number of solutions for each case is as follows: one-loop (triangle), 2; two-loops (pentad), 8; three-loops type 3(a,b,c), 16, 24, and 32, respectively. For the topologically similar planar mechanisms, these numbers are 2, 6, 14, 16, and 18. It is curious that the complexity of the three-loop spherical mechanisms is more strongly affected by topology than their planar counterparts. As shown by examples in the Appendix, spherical structures with all real roots are known for the triangle, the pentad and structure 3a, but it is not yet known whether there exist structures of type 3b or 3c having all real roots.

These results have application to forward and inverse kinematic solutions for all kinds of spherical mechanisms, from one-degree-of-freedom mechanisms up to multi-degree-of-freedom robots. The methods also apply to solving the rotational component of the loop closure equations for spatial mechanisms. Also, the elimination approach introduced here may have uses in the treatment of other spatial mechanisms and systems of polynomial equations arising in other fields.

Nomenclature

$A \cup B =$ union of sets A and B

$A \otimes B =$ product of sets: {all $x | x = ab$, $a \in A$, $b \in B$}

$A \backslash B =$ set minus: $A - (A \cap B)$

$c_i, s_i, t_i =$ half-angle functions $\cos(\theta_i/2), \sin(\theta_i/2), \tan(\theta_i/2)$

Appendix A

Numerical Examples. This appendix tabulates an example problem and its solution for each mechanism type. Side rotations are given in terms of rotation functions $R_x(\alpha)$ and $R_z(\alpha)$, which are right-handed rotations of angle α (in radians) about the x-axis and z-axis, respectively. To save space, answers are given as tangent-half-angles for joints 1, 2, and 3. Even though there are nine joints in the three-loop mechanisms, they have been numbered so that the first three determine the relative orientation of all the ternary and quaternary links, hence locating all the joint axes. All computations were done in double precision using Matlab. After joint values were computed from the eigenvectors, the residuals in the loop equations were checked. The magnitude of the residuals was always better than 10^{-7}, and commonly better than 10^{-12}, showing a high level of numerical stability. Of course, random sampling does not encounter problems having multiple roots or other singularities, which might require more specialized procedures to compute accurately.

Triangle. The given sides (units in radian) are

$$S_1 = R_x(.3), \quad S_2 = R_x(.4), \quad S_3 = R_x(.5).$$

The numerical solutions are listed in Table 1.

Pentad. The given sides (units in radian) are

$$S_1 = R_x(2.09), \quad S_2 = R_x(4.59), \quad S_3 = R_x(5.24),$$

$$S_4 = R_x(4.84), \quad S_5 = R_z(4.98)R_x(4.22),$$

$$S_6 = R_z(2.15)R_x(4.59), \quad S_7 = R_x(1.42).$$

This example is notable for having all real roots. The numerical solutions are listed in Table 2.

Table 1 Solutions for triangle (tangent half-angles)

	t_1	t_2	t_3
1	−1.949937	−0.979864	−2.900527
2	1.949937	0.979864	2.900527

Type 3a. The given sides (units in radian) are:

$$S_1 = R_x(4.863), \quad S_2 = R_z(1.029)R_x(5.339),$$

$$S_3 = S_2'S_1', \quad S_4 = R_z(0.893)R_x(1.857),$$

$$S_5 = R_z(5.464)R_x(1.655),$$

$$S_6 = R_z(5.884)R_x(1.448),$$

Table 2 Solutions for pentad (tangent half-angles)

	t_1	t_2	t_3
1	7.791279	−0.361058	0.107830
2	−2.294342	1.330759	−0.429469
3	2.005683	0.096003	−0.492788
4	−0.265766	−0.785437	3.773640
5	0.042981	5.488394	−29.689637
6	0.258660	−12.402011	3.878452
7	1.228881	0.164803	−0.809339
8	1.084466	−2.835662	0.917918

Table 3 Solutions for type 3a (tangent half-angles)

	t_1	t_2	t_3
1	−48.566592	−2.515025	0.061630
2	42.882497	1.034898	−5.306974
3	9.041544	−3.058067	0.074276
4	2.432176	0.600422	−1.445874
5	2.571563	−5.427725	−1.764800
6	−2.038693	−1.152379	−3.777007
7	1.602743	0.468618	2.361986
8	−1.567532	−10.216266	0.073543
9	−1.318230	−5.005893	0.083233
10	−1.003469	−3.087755	−1.992006
11	1.034471	5.927095	−0.076588
12	−0.638457	−0.030277	−0.903191
13	0.421327	−0.314664	4.672430
14	0.373999	0.787632	4.480535
15	−0.066276	0.342857	2.205674
16	−0.053188	−1.157316	−0.097254

Table 4 Solutions for type 3b (tangent half-angles)

	t_1	t_2	t_3
1	−0.300238	−0.975914	−0.141182
	8.099887i	1.285612i	0.750772i
3	−4.631640	−0.558862	−0.826819
4	5.155489	5.750268	0.839257
5	−3.222447	−0.358801	2.598104
6	−0.014653	0.125571	0.011801
	3.010856i	0.864406i	−1.052918i
8	3.491401	2.753540	−2.720061
9	2.801305	−2.288869	−1.808784
10	1.758171	−2.351800	1.824165
11	1.342918	1.194637	−2.074716
12	1.021104	1.082175	0.983192
13	−0.911262	0.029402	−1.012938
14	−0.722349	0.041045	1.892119
15	0.535529	−2.745870	1.627629
16	0.336466	−3.201457	−2.252201
17	−0.337104	−0.063246	2.073473
18	−0.000938	0.125950	0.013773
	0.127673i	−0.864050i	1.061203i
20	−0.008712	−1.048319	−0.214501
	0.042197i	1.256272i	0.856293i
22	−0.175480	−0.413658	−0.831313
23	−0.148142	−0.676910	−0.840920
24	−0.149006	−0.936849	3.028856

Table 5 Solutions for type 3c (tangent half-angles)

	t_1	t_2	t_3
1	−10.949013	−1.280400	0.021278
2	5.010578	0.687520	7.722941
3	4.539627	−1.383537	−2.719863
4	4.051448	0.746949	−0.155187
5	−2.337324	−1.232473	−3.300713
	2.691620i	0.291555i	−3.383834i
7	3.132934	0.838985	4.821763
8	−2.312559	−0.771448	−0.086123
9	−2.144114	−0.037149	−0.654865
10	−1.804988	−0.200343	1.826531
11	1.502078	1.221439	−0.272128
12	0.585683	0.134565	−0.386102
	1.182659i	−4.205445i	0.834128i
14	0.645892	1.826968	0.624237
	1.090596i	−2.428202i	1.068532i
16	1.040232	−0.736672	0.331547
17	−0.379239	0.038893	−1.491983
	0.964691i	−0.485738i	−0.192974i
19	−0.227070	0.024609	0.524896
	0.956013i	−0.513776i	−0.613175i
21	0.262808	2.790265	0.583044
	0.926871i	3.143888i	0.821044i
23	0.415773	2.574668	−0.686235
	0.761260i	0.876722i	0.290866i
25	0.380018	1.136403	−0.309627
26	0.321539	−0.181036	2.211928
27	0.029948	0.027042	−0.070381
	0.195819i	−0.287436i	−0.140097i
29	−0.123758	0.124726	4.655016
	0.117978i	−0.388148i	9.603347i
31	0.066038	0.477103	4.190505
32	0.055442	0.250194	−0.154052

$$S_7 = R_x(1.454), \quad S_8 = R_x(1.530),$$

$$S_9 = R_x(5.383), \quad S_{10} = R_x(1.739),$$

$$S_{11} = R_x(1.950), \quad S_{12} = R_x(5.088).$$

This example, found after approximately one million random trials, has 16 real roots, listed in Table 3. About 80% of random examples have no real roots.

Type 3b. The given sides (units in radian) are:

$$S_1 = R_x(1.76), \quad S_2 = R_z(2.30)R_x(1.46),$$

$$S_3 = R_x(4.27), \quad S_4 = R_x(1.20),$$

$$S_5 = R_z(0.81)R_x(0.41), \quad S_6 = R_x(5.03),$$

$$S_7 = R_x(1.49), \quad S_8 = R_x(0.87),$$

$$S_9 = R_z(0.36)R_x(0.11), \quad S_{10} = R_x(4.77),$$

$$S_{11} = R_x(4.01)R_z(0.88).$$

This example has 16 real and 8 complex roots, listed in Table 4. The complex roots occupy two rows each, with the imaginary part appearing in the second row. The complex conjugate root is not written in the table, but is implied; the numbering in the first column reflects this.

Type 3c. The given sides (units in radian) are:

$$S_1 = R_x(5.01), \quad S_2 = R_x(5.59),$$

$$S_3 = R_x(1.39), \quad S_4 = R_x(3.76)R_z(1.00),$$

$$S_5 = R_z(0.24)R_x(1.33), \quad S_6 = R_x(1.78),$$

$$S_7 = R_x(4.82), \quad S_8 = R_x(2.74)R_z(1.76),$$

$$S_9 = R_z(1.66)R_x(1.16), \quad S_{10} = R_x(4.61),$$

$$S_{11} = R_x(4.74).$$

This type 3c example has 14 real and 18 complex roots. Since the planar structure of this type can have 18 real roots, at least that many must be possible for the spherical case, but 14 real is the most found in 100,000 random trials. In Table 5, as in the prior example, the complex roots occupy two rows each, with the imaginary part appearing in the second row. The complex conjugate root is not written in the table, but is implied; the numbering in the first column reflects this.

References

[1] Nielsen, J., and Roth, B., 1999, "Solving the Input/Output Problem for Planar Mechanisms," ASME J. Mech. Des., **121**(2), pp. 206–211.

[2] Wampler, C. W., 1999, "Solving the Kinematics of Planar Mechanisms," ASME J. Mech. Des., **121**(3), pp. 387–391.

[3] Wampler, C. W., 2001, "Solving the Kinematics of Planar Mechanisms by Dixon Determinant and a Complex-Plane Formulation," ASME J. Mech. Des., **123**(3), pp. 382–387.

[4] Duffy, J., 1980, *Analysis of Mechanisms and Robot Manipulators*, Halsted Press, John Wiley and Sons, New York.

[5] Kline, M., 1972, *Mathematical Thought From Ancient to Modern Times*, Vol. 1, Oxford University Press, Oxford, UK.

[6] Gosselin, C., Sefrioui, J., and Richard, M. J., 1994, "On the Direct Kinematics of Spherical Three-Degree-of-Freedom Parallel Manipulators of General Architecture," ASME J. Mech. Des., **116**(2), pp. 594–598.

[7] Innocenti, C., and Parenti-Castelli, V., 1993, "Echelon Form Solution of Direct Kinematics for the General Fully-Parallel Spherical Wrist," Mech. Mach. Theory, **28**(4), pp. 553–561.

[8] Wohlhart, K., 1994, "Displacement Analysis of the General Spherical Stewart Platform," ASME J. Mech. Des., **29**(4), pp. 581–589.

[9] Dhingra, A. K., Almadi, A. N., and Kohli, D., 2000, "Closed-Form Displacement Analysis of 8, 9 and 10-Link Mechanisms. Part I: 8-Link 1-DOF Mechanisms," Mech. Mach. Theory, **35**(6), pp. 821–850.

[10] Larochelle, P., and McCarthy, J. M., 1995, "Planar Motion Synthesis Using an Approximate Bi-invariant Metric," ASME J. Mech. Des., **117**(4), pp. 646–651.

[11] Manolescu, N. I., 1973, "A Method Based on Baranov Trusses,and Using Graph Theory to Find the Set of Planar Jointed Kinematic Chains and Mechanisms," Mech. Mach. Theory, **8**, pp. 3–22.

[12] Innocenti, C., 1995, "Polynomial Solution to the Position Analysis of the 7-Link Assur Kinematic Chain With One Quaternary Link," Mech. Mach. Theory, **30**(8), pp. 1295–1303.

[13] Shen, H., Ting, K.-L., and Yang, T., 2000, "Configuration Analysis of Complex Multiloop Linkages and Manipulators," Mech. Mach. Theory, **35**, pp. 353–362.

[14] Raghavan, M., and Roth, B., 1995, "Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators," ASME J. Mech. Des., **117**(B), pp. 71–79.

[15] Dhingra, A. K., Almadi, A. N., and Kohli, D., 2000, "A Gröbner-Sylvester Hybrid Method for Closed-Form Displacement Analysis of Mechanisms," ASME J. Mech. Des., **122**(4), pp. 431–438.

[16] Cox, D., Little, J., and O'Shea, D., 1992, *Ideals, Varieties, and Algorithms*, Springer-Verlag, New York.

[17] Auzinger, W., and Stetter, H. J., 1988, "An Elimination Algorithm for the Computation of All Zeros of a System of Multivariate Polynomial Equations," in *Conf. in Numerical Analysis, ISNM, Vol. 86*, Birkhäuser Verlag, pp. 11–30.

[18] Möller, H. M., and Stetter, H. J., 1995, "Matrix Polynomial Equations With Multiple Zeros Solved by Matrix Eigenproblems," Numer. Math., **70**, pp. 311–329.

[19] Morgan, A. P., and Sommese, A. J., 1987, "A Homotopy for Solving General Polynomial Systems That Respects *m*-Homogeneous Structures," Appl. Math. Comput., **24**, pp. 101–113.

[20] Moler, C. B., and Stewart, G. W., 1973, "An Algorithm for Generalized Matrix Eigenvalue Problems," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **10**(2), pp. 241–256.