PERGAMON

Mechanism and Machine Theory 36 (2001) 1121–1134

MECHANISM AND MACHINE THEORY

www.elsevier.com/locate/mechmt

Direct kinematic analysis of 3-RS parallel mechanisms

Jinwook Kim, F.C. Park *

School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Republic of Korea

Received 28 February 2000; received in revised form 24 April 2001; accepted 10 May 2001

Abstract

This article presents a direct kinematic analysis of 3-RS parallel mechanisms. A 3-RS parallel mechanism consists of a fixed base and a moving platform connected by three serial chains, with each serial chain containing one passive revolute joint and one passive spherical joint. A variety of 3-RS mechanisms have been proposed in the literature, to overcome the disadvantages of small workspace characteristic of Stewart platforms and other 6–6 parallel mechanisms. We provide a computationally efficient method to solve the direct kinematics problem for general 3-RS mechanisms. By appealing to Sylvester's dialytic elimination method, the direct kinematics problem can be reduced to the solution of a 16th order polynomial equation in a single variable. In our approach the polynomial coefficients are represented in terms of convolutions of vectors, which considerably simplifies the coding of the algorithm, and unlike existing approaches does not rely on symbolic computation software. The method is applied to the direct kinematic analysis of the Eclipse, a novel six d.o.f. 3-RS parallel mechanism designed for five-face machining.

1. Introduction

Beginning with the six-linear jack system of Gough and Whitehall [1] and the flight simulator platform devised by Stewart [2], parallel mechanisms are today being used in a wide variety of applications, ranging from machine tool platforms (e.g, [3] and the references cited therein) to robot manipulators (e.g., [4]). A typical six degree-of-freedom (d.o.f.) parallel mechanism consists of a moving platform connected to a fixed base by six serial limbs. Each limb is driven by a single actuator, and all the actuators are mounted on or nearby the fixed base. Due to its parallel structure and the low inertia of the moving parts, these parallel mechanisms offer the advantages of higher overall stiffness, lower inertia, and higher operating speeds. These advantages however

* Corresponding author. Tel.: +82-2-880-7133; fax: +82-2-883-1513.
E-mail address: fcp@plaza.snu.ac.kr (F.C. Park).

0094-114X/01/$ - see front matter
PII: S0094-114X(01)00042-8

come at the expense of a reduced workspace, difficult mechanical design, and more complex kinematics and control algorithms.

To overcome the shortcomings associated with six-limb platforms, various alternative architectures possessing fewer limbs have been investigated in the literature (see Fig. 1). In general, it has been observed that by reducing the number of serial chains connected in parallel, it is possible to enlarge the workspace of the parallel mechanism, although at a cost in reduced stiffness and speed due to the hybrid serial-parallel structure, and the increased inertia due to the relocation of the actuators.

Behi [5] proposes a six d.o.f. parallel mechanism consisting of three PRPS serial structures, with the two prismatic joints actuated (here P denotes a prismatic joint, R denotes a revolute joint, and

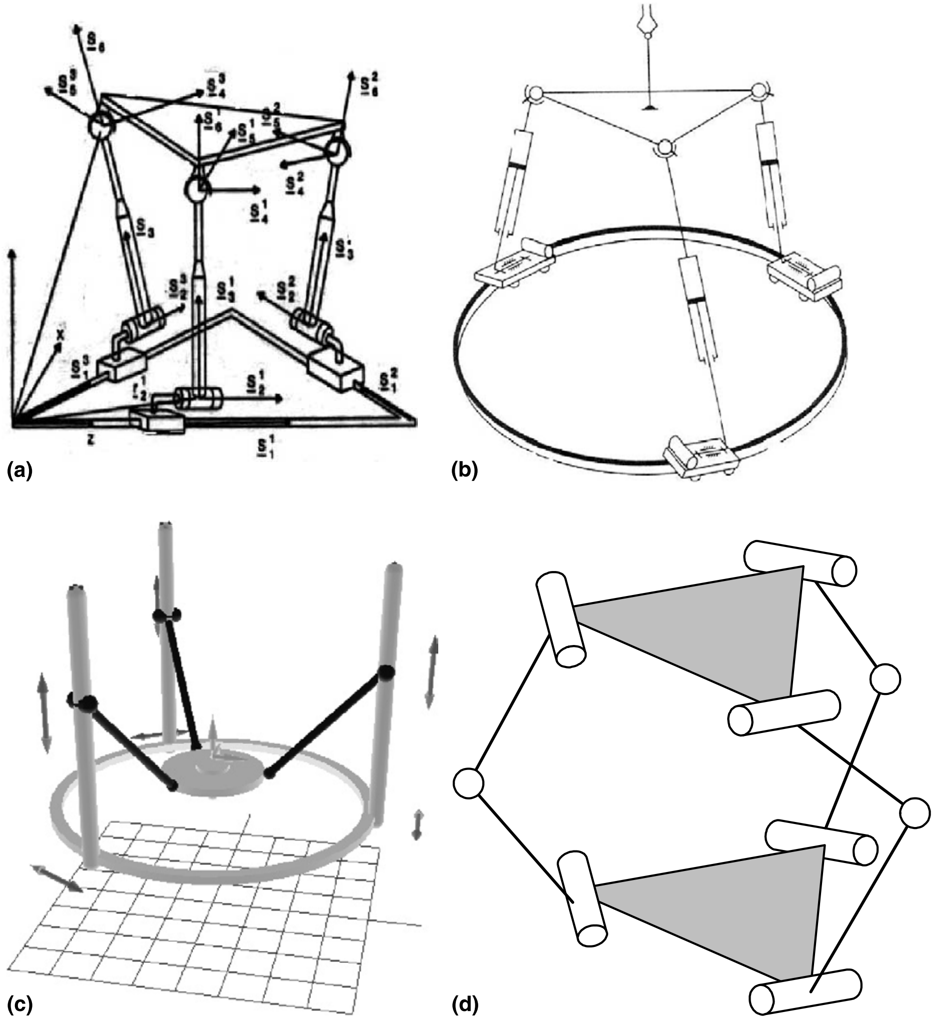

Fig. 1. Examples of 3-RS mechanisms: (a) the Behi Mechanism [5]; (b) the Alizade mechanism [6]; (c) the Eclipse mechanism [7]; and (d) the Canterbury Tracker Mechanism [8].

S denotes a spherical, or ball, joint). Alizade et al. [6] propose a similar topologically equivalent six d.o.f. mechanism, in which the base of the three serial limbs slide along a circular rather than a linear guideway. The Eclipse mechanism proposed by Kim et al. [7] consists of three PPRS serial structures that also slide along a circular guideway. Dunlop [8] also examines the kinematics of a 3-RS mechanism, called the Canterbury tracker, that was originally proposed by Hunt.

What the above mechanisms share in common is that they all consist of three serial structures connected in parallel, with each serial chain containing exactly one passive revolute and one passive spherical joint; for this reason we refer to this class of mechanisms as 3-RS mechanisms. When all the actuators are locked, these mechanisms become topologically equivalent.

The objective of this paper is the efficient analysis of the direct (or forward) kinematics of 3-RS mechanisms. Direct kinematic analysis is an essential component of the design, programming and control of any mechanism. Due to its difficulty, the direct kinematics problem for parallel mechanisms has received extensive attention in the literature. The drawback with standard numerical approaches like the Newton–Raphson method is that, although they are reliable and efficient, in general only one solution can be found for each initial guess, and in the case of kinematic singularities may fail to converge. Particularly at the design stage, efficient and failure-free methods of direct kinematic analysis that are guaranteed to find all the solutions are essential.

Recently Sylvester's dialytic elimination method has been effectively used to solve the direct kinematics of many special parallel mechanism designs (see [9] for a discussion and literature survey). Nanua et al. [10] offer a solution to the direct kinematics of a special symmetric Stewart platform, while Husty [11] provides a remarkable solution for the general 6–6 Stewart platform. Dunlop and Jones [8] also offer a dialytic elimination-based solution for the three d.o.f. Canterbury Tracker mechanism (see Fig. 1).

In this paper we develop a computationally efficient method to solve the direct kinematics for the general class of 3-RS parallel mechanisms. We show how to reduce the direct kinematics equations to a system of polynomials, and apply Sylvester's dialytic elimination method to further reduce this system to a single 16th order polynomial equation in one variable.

One of the novel features of our approach is that the coefficients of the polynomials are represented in terms of convolutions of vectors. This considerably simplifies the coding of the algorithm. In particular, there is no need to rely on symbolic computation software to compute the coefficients during the elimination process. Another motivation of our paper is to introduce the Eclipse [7], a novel six d.o.f. 3-RS mechanism with a significantly enhanced workspace that is capable of continuously accessing five faces of a cubic workpiece. We apply our algorithm to evaluate all solutions to the direct kinematics of the Eclipse.

2. Closure equations

Each of the three serial chains in a 3-RS parallel mechanism contains one passive revolute and one passive spherical joint. Assuming all the actuated joints are locked in place, Fig. 2 shows the generally topology of a 3-RS parallel mechanism. It is not difficult to see that all of the previous examples of 3-RS parallel mechanisms can be reduced to this equivalent topology.

In what follows we formulate the closure equations under the assumption that aside from the R and S joint in each serial chain, all the other joints are locked. First, suppose a fixed frame has

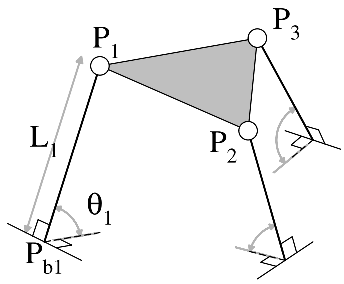

Fig. 2. The topology of a 3-RS mechanism with all actuated joints locked.

been chosen, and let P_{b1}, P_{b2}, P_{b3} denote the locations of the revolute joints with respect to the fixed frame origin. Let L_1, L_2, L_3 denote the lengths of the fixed links, and let P_1, P_2, P_3 denote the positions of the three spherical joints with respect to the fixed frame. For each serial subchain, define the angles θ_i and ϕ_i, $i = 1,2,3$ with respect to the fixed frame as shown in Fig. 3. Then from the requirement that the distances between the ball joints of the moving platform are fixed, the kinematic constraint equations can be expressed in the form $g{:}R^3 \to R^3$, where

$$g(\theta) = \frac{1}{2}\begin{bmatrix} \|P_1 - P_2\|^2 - D^2 \\ \|P_2 - P_3\|^2 - D^2 \\ \|P_3 - P_1\|^2 - D^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{1}$$

$$P_i = P_{bi} + L_i \begin{bmatrix} \cos(\phi_i)\cos(\theta_i) \\ \sin(\phi_i)\cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}. \tag{2}$$

Generally P_{bi}, L_i, and ϕ_i, $i = 1,2,3$, are functions of the actuated joint values and the geometric parameters of the mechanism, and can be assumed known. The direct kinematic problem is to find all the solutions θ to the constraint equation (1).

The constraint equation (1) can be expanded as

$$g_i(\theta) = k_{i1} + k_{i2}\cos\theta_i + k_{i3}\cos\theta_{i+1} + k_{i4}\sin\theta_i + k_{i5}\sin\theta_{i+1} + k_{i6}\cos\theta_i\cos\theta_{i+1} + k_{i7}\sin\theta_i\sin\theta_{i+1},$$

where $i = 1,2,3$, all the subscripts of θ are moduli of 3, and the coefficients k_{ij} are given in the Appendix A.

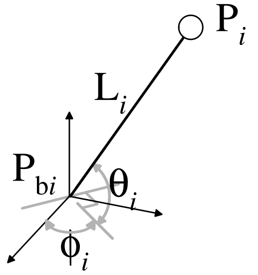

Fig. 3. Definition of the angles θ_i and ϕ_i.

We now adopt the following standard trigonometric substitutions used in many kinematics problems:

$$t_i = \tan\frac{\theta_i}{2},$$
$$\cos\theta_i = \frac{1-t_i^2}{1+t_i^2},$$
$$\sin\theta_i = \frac{2t_i}{1+t_i^2}.$$

The constraint equations can now be expressed in terms of the new variables t_1, t_2, t_3 as follows:

$$g_1 = \begin{bmatrix} t_1^2 & t_1 & 1 \end{bmatrix} A \begin{bmatrix} t_2^2 \\ t_2 \\ 1 \end{bmatrix}, \tag{3}$$

$$g_2 = \begin{bmatrix} t_2^2 & t_2 & 1 \end{bmatrix} B \begin{bmatrix} t_3^2 \\ t_3 \\ 1 \end{bmatrix}, \tag{4}$$

$$g_3 = \begin{bmatrix} t_3^2 & t_3 & 1 \end{bmatrix} C \begin{bmatrix} t_1^2 \\ t_1 \\ 1 \end{bmatrix}, \tag{5}$$

where the matrices A, B, C are given explicitly in the Appendix B. In Section 3 we discuss how to systematically solve the system of cyclic polynomials given above.

3. Direct kinematic analysis

We now describe an algebraic method for solving the constraint equation (1) based on dialytic elimination. Observe that Eq. (1) consists of three nonlinear equations in three unknowns. Using dialytic elimination, we eliminate two of the three variables, which results in a single 16th order polynomial equation in one unknown. Obtaining the solutions of such polynomials is by now quite standard. In what follows all matrices are represented in uppercase, and vectors are aligned in column order and represented in lowercase.

In order to eliminate variable t_3 from (4) and (5), rearrange (5) as follows:

$$g_3' = \begin{bmatrix} t_1^2 & t_1 & 1 \end{bmatrix} C^{\mathrm{T}} \begin{bmatrix} t_3^2 \\ t_3 \\ 1 \end{bmatrix}. \tag{6}$$

Let $\beta_i(t_2)$ and $\gamma_i(t_1)$, respectively, denote the ith columns of matrices B and C^{T}, i.e.,

$$\beta_i(t_2) = \begin{bmatrix} t_2^2 & t_2 & 1 \end{bmatrix} b_i,$$
$$\gamma_i(t_1) = \begin{bmatrix} t_1^2 & t_1 & 1 \end{bmatrix} c_i.$$

Then (4) and (6) can be re-expressed as follows:

$$g_2 = \beta_1 t_3^2 + \beta_2 t_3 + \beta_3,$$
$$g_3' = \gamma_1 t_3^2 + \gamma_2 t_3 + \gamma_3. \tag{7}$$

In order for (7) to have a solution in terms of the variable t_3, the following must hold:

$$\det(R_1) = 0, \quad R_1 = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & 0 \\ 0 & \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 & 0 \\ 0 & \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}$$

Observe furthermore that $\det(R_1)$ can be calculated as $\det(R_1) = \sum_{ijkl}\omega_{ijkl}\beta_i\beta_j\gamma_k\gamma_l$, where the coefficients ω are given in the Appendix C. Also observe that since β_i is a second-order polynomial in t_2, $\beta_i\beta_j$ is a fourth-order polynomial in t_2. Denoting $\beta_i\beta_j$ by

$$\beta_i\beta_j = \begin{bmatrix} t_2^4 \\ t_2^3 \\ t_2^2 \\ t_2 \\ 1 \end{bmatrix} b_{ij},$$

the relation between b_i, b_j, and b_{ij} is given by

$$b_{ij} = b_i * b_j,$$

where (*) denotes convolution of vectors, i.e., if $c = a*b$ is the convolution between an m-dimensional vector a and an n-dimensional vector b, then the components of c are given by

$$c_i = \sum_{j=\max(1,i+1-n)}^{\min(i,m)} a_j b_{i-j+1},$$

where a_i, b_j, c_k are, respectively, the ith, jth, and kth elements of a, b, c.

Based on the above, the equation $\det(R_1) = 0$ can be reduced to

$$g_4 = \begin{bmatrix} t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \end{bmatrix} D \begin{bmatrix} t_2^4 \\ t_2^3 \\ t_2^2 \\ t_2 \\ 1 \end{bmatrix} = 0, \tag{8}$$

where

$$D = \sum_{ijkl} \omega_{ijkl}(c_k * c_l)(b_i * b_j)^{\mathrm{T}}. \tag{9}$$

The requirement that (3) and (8), which have only t_1 and t_2 as variables, have a solution is as follows:

$$\det(R_2) = 0, \quad R_2 = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5 & 0 \\ 0 & \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5 \\ \alpha_1 & \alpha_2 & \alpha_3 & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix},$$

where

$$\delta_i = \begin{bmatrix} t_1^4 \\ t_1^3 \\ t_1^2 \\ t_1 \\ 1 \end{bmatrix} d_i$$

and d_i denotes the ith column of D. Note that

$$\det(R_2) = \sum_{ijklmn} \lambda_{ijklmn} \delta_i \delta_j \alpha_k \alpha_l \alpha_m \alpha_n.$$

Consult the Appendix D for the exact values of λ.

Eliminating the coefficients t_2, t_3 from (3)–(5), the 16th-order polynomial in t_1 can now be obtained by setting $\det(R_2) = 0$:

$$\det(R_2) = \sum_{i=1}^{17} e_i t_1^{17-i}, \tag{10}$$

$$e = \sum_{ijklmn} \lambda_{ijklmn} d_i * d_j * a_k * a_l * a_m * a_n. \tag{11}$$

There exist many methods for finding the roots of a polynomial, e.g., eigenvalue methods for the companion matrix constructed with the coefficients e_i. Among the 16 solutions, only real roots are physically meaningful. Finding all the roots of a polynomial can also be computationally expensive, and floating point overflow may occur due to the high order of the resultant. One means of alleviating this is to normalize the coefficients; in our case the norm of the coefficient matrices A, B, C in Eqs. (3)–(5) is normalized to be close to one (see [9] and the references cited for a further discussion on numerical issues). Once t_1 has been obtained, t_2 and t_3 can be solved by substituting t_1 into (3) and (5), resulting in second-order polynomials. Values for the passive joints can be calculated from the relation

$$\theta_{i+6} = 2\tan^{-1}(t_i).$$

From the obtained values of the active and passive joints, the position and orientation of the tool frame can be found.

In the Appendix E, matlab code for solving the system of polynomials (3)–(5) is available.

4. Case study: the Eclipse

In this section we apply the above solution technique to the direct kinematic analysis of the Eclipse mechanism. As shown earlier in Fig. 1, the Eclipse consists of three PPRS serial subchains (P here denotes prismatic, or sliding joint), with the first P joint denoting sliding motion along the circular guideway. The mechanism has six kinematic degrees of freedom, with the first two prismatic joints actuated (indicated by arrows).

A hardware prototype of the Eclipse mechanism is shown in Fig. 4. For our analysis we set the radii of the circular guideway and moving platform to be 300 and 100 mm, respectively. The length of each of the links connecting the vertical columns to the moving platform is 330 mm. The prototype is intended for five-face machining, and is capable of accessing the five visible faces of a cubic workpiece.

For the direct kinematic analysis it is sufficient to consider the nine joints excluding the three spherical joints. Denote the six actuated joints by (ϕ_1, ϕ_2, ϕ_3, h_1, h_2, h_3) and the three passive joints by (θ_1, θ_2, θ_3). In the constraint equation (1), P_{bi} for the Eclipse is given as

$$P_{bi} = [R\cos\phi_i \quad R\sin\phi_i \quad h_i]^{\mathrm{T}}.$$

Following the procedure outlined in Section 3, we can derive the 16th-order polynomial equation in a single unknown, whose solutions provide all possible configurations of the end-effector for a given set of actuated joint values.

Fig. 4. The Eclipse prototype.

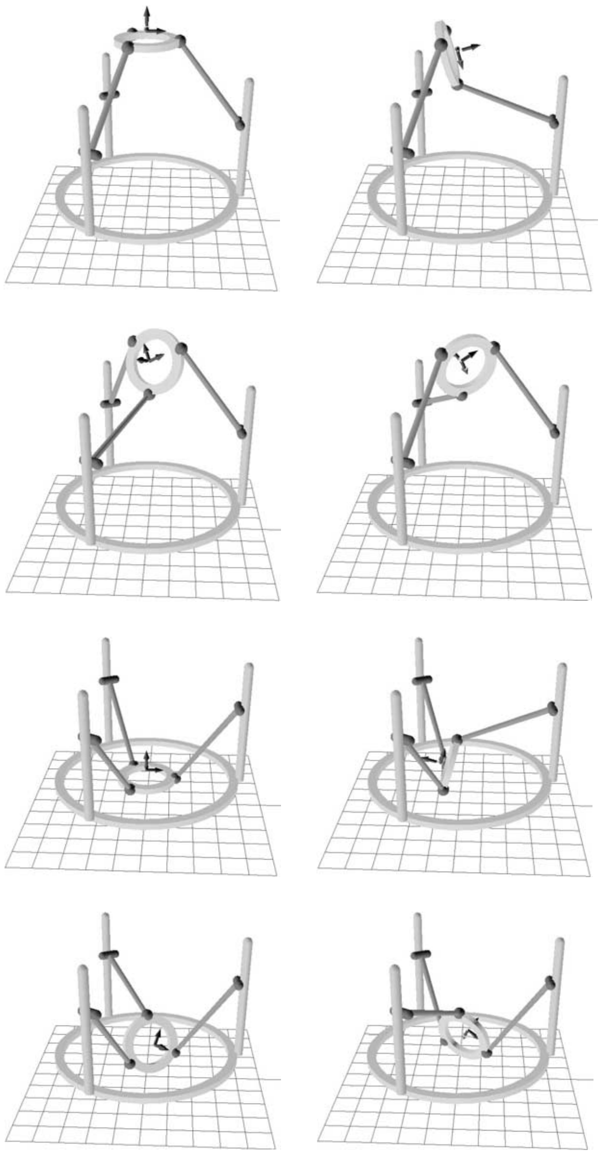

Fig. 5. The eight possible solutions corresponding to the home configuration.

For the set of actuated joint values corresponding to the home position (i.e., when the spindle is aligned vertically over the center of the workspace), the Eclipse has eight real solutions; the configurations corresponding to the real solutions are shown in Fig. 5.

5. Conclusion

In this paper we have applied Sylvester's dialytic elimination method to solve the direct kinematics of a general class of parallel mechanisms. These mechanisms, because they consist of three serial subchains connecting two platforms, with each subchain containing one passive revolute and one passive spherical joint, are referred to as 3-RS mechanisms. We develop a computationally efficient procedure in which the kinematic constraint equations are reduced to a single 16th-order polynomial in a single unknown. We then perform a direct kinematic analysis of a novel six d.o.f. parallel mechanism capable of five-face machining, the Eclipse. One of the open questions raised by this research is whether in fact a 3-RS mechanism with 16 real solutions can be realized; simulation studies of the Eclipse and other similar 3-RS mechanisms indicate an upper bound of eight real solutions. The approach of Dietmaier [12] could provide a definitive answer to this question.

Acknowledgements

This research was supported by a grant from the National Research Lab for CAD/CAM, and by the BK_{21} program in Mechanical Engineering at Seoul National University.

Appendix A. Coefficients p_{ij}

$$\begin{aligned}
k_{i1} &= \|P_{bi} - P_{bi+1}\|^2 + L_i^2 + L_{i+1}^2, \\
k_{i2} &= 2L_i(P_{bi_x} - P_{bi+1_x})(\cos\phi_i + \sin\phi_i), \\
k_{i3} &= -2L_{i+1}(P_{bi_x} - P_{bi+1_x})(\cos\phi_{i+1} + \sin\phi_{i+1}), \\
k_{i4} &= 2L_i(P_{bi_z} - P_{bi+1_z}), \\
k_{i5} &= -2L_{i+1}(P_{bi_z} - P_{bi+1_z}), \\
k_{i6} &= -2L_iL_{i+1}\cos(\phi_i - \phi_{i+1}), \\
k_{i7} &= -2L_iL_{i+1}.
\end{aligned}$$

Appendix B. Matrix A, B, C

$$A = \begin{bmatrix} k_{11} - k_{12} - k_{13} + k_{16} & 2k_{15} & k_{11} - k_{12} + k_{13} - k_{16} \\ 2k_{14} & 4k_{17} & 2k_{14} \\ k_{11} + k_{12} - k_{13} - k_{16} & 2k_{15} & k_{11} + k_{12} + k_{13} + k_{16} \end{bmatrix},$$

$$B = \begin{bmatrix} k_{21} - k_{22} - k_{23} + k_{26} & 2k_{25} & k_{21} - k_{22} + k_{23} - k_{26} \\ 2k_{24} & 4k_{27} & 2k_{24} \\ k_{21} + k_{22} - k_{23} - k_{26} & 2k_{25} & k_{21} + k_{22} + k_{23} + k_{26} \end{bmatrix},$$

$$C = \begin{bmatrix} k_{31} - k_{32} - k_{33} + k_{36} & 2k_{35} & k_{31} - k_{32} + k_{33} - k_{36} \\ 2k_{34} & 4k_{37} & 2k_{34} \\ k_{31} + k_{32} - k_{33} - k_{36} & 2k_{35} & k_{31} + k_{32} + k_{33} + k_{36} \end{bmatrix}.$$

Appendix C. Non-zero elements of ω

i	j	k	l	ω_{ijkl}
1	1	3	3	1
1	2	2	3	−1
1	3	2	2	1
1	3	1	3	−2
2	2	1	3	1
2	3	1	2	−1
3	3	1	1	1

Appendix D. Non-zero elements of λ

i	j	k	l	m	n	λ_{ijklmn}	i	j	k	l	m	n	λ_{ijklmn}
5	5	1	1	1	1	1	1	5	1	2	2	3	−4
3	5	1	1	1	3	−2	1	3	1	3	3	3	−2
4	5	1	1	1	2	−1	2	2	1	3	3	3	1
4	4	1	1	1	3	1	2	5	1	2	2	2	−1
1	5	1	1	3	3	2	2	4	1	2	2	3	1
3	5	1	1	2	2	1	2	3	1	2	3	3	−1
2	4	1	1	3	3	−2	1	1	3	3	3	3	1
3	4	1	1	2	3	−1	1	5	2	2	2	2	1
2	5	1	1	2	3	3	1	2	2	3	3	3	−1
3	3	1	1	3	3	1	1	3	2	2	3	3	1
1	4	1	2	3	3	3	1	4	2	2	2	3	−1

Appendix E. Matlab code for solving Eqs. (3)–(5)

```
function [t, error] = roots32(a, b, c)
% ROOTS32  Find roots of system of polynomial
%
% [t, error] = root32(A, B, C)
%
% Inputs A, B, C are matrices consist of coefficients of equations.
%
% [t1^2 t1 1] A [t2^2; t2; 1] = 0
```

```
% [t2^2 t2 1] B [t3^2; t3; 1] = 0
% [t3^2 t3 1] C [t1^2; t1; 1] = 0
%
% Output t are roots of equations.
%
% t(i,:) = [t1 t2 t3]
%
% If you want to check how much its roots are exact, see value of 'error'.

eps = 1e-5;

g4 = [1 1 1 3 3; -1 1 2 2 3; 1 1 3 2 2; -2 1 3 1 3; 1 2 2 1 3; -1 2 3 1 2; 1 3 3 1 1];
g5 = [...
+1,5,5,1,1,1,1;   -2,3,5,1,1,1,3;   -1,4,5,1,1,1,2;   +1,4,4,1,1,1,3;
+2,1,5,1,1,3,3;   +1,3,5,1,1,2,2;   -2,2,4,1,1,3,3;   -1,3,4,1,1,2,3;
+3,2,5,1,1,2,3;   +1,3,3,1,1,3,3;   +3,1,4,1,2,3,3;   -4,1,5,1,2,2,3;
-2,1,3,1,3,3,3;   +1,2,2,1,3,3,3;   -1,2,5,1,2,2,2;   +1,2,4,1,2,2,3;
-1,2,3,1,2,3,3;   +1,1,1,3,3,3,3;   +1,1,5,2,2,2,2;   -1,1,2,2,3,3,3;
+1,1,3,2,2,3,3; -1,1,4,2,2,2,3];

c = c';

% constructing coefficient matrix D
d = zeros(5,5);
for i = 1:7
   bij = conv(b(:,g4(i,2)),b(:,g4(i,3)));
   ckl = conv(c(:,g4(i,4)),c(:,g4(i,5)));
   d = d+g4(i,1)*ckl*bij';
end

% constructing resultant coefficient vector e
e = zeros(17,1);
for p = 1:22
   cof = g5(p,1);   i = g5(p,2);   j = g5(p,3);   k = g5(p,4);   l = g5(p,5);
   m = g5(p,6); n = g5(p,7);
   e = e+cof*conv(conv(conv(conv(conv(d(:,i),d(:,j)),a(:,k)),
   a(:,l)),a(:,m)),a(:,n));
end

% solving resultant
t1 = roots(e);

% compute all roots of equations
t = zeros(16, 3);
t(:,1) = t1;
isfirst = 1;
for i = 1:16
```

```
    if (isfirst)
      t2=roots([t1(i)^2 t1(i) 1]*a);
      t3=roots([t1(i)^2 t1(i) 1]*c).';

      s=[t2.^2 t2 [1;1]]*b*[t3.^2; t3; 1 1];
      flag=1;
      if (flag & abs(s(1,1))<eps) t(i,2)=t2(1); t(i,3)=t3(1); flag=0;
    end;
      if (flag & abs(s(1,2))<eps) t(i,2)=t2(1); t(i,3)=t3(2); flag=0;
    end;
      if (flag & abs(s(2,1))<eps) t(i,2)=t2(2); t(i,3)=t3(1); flag=0;
    end;
      if (flag & abs(s(2,2))<eps) t(i,2)=t2(2); t(i,3)=t3(2); flag=0;
    end;

      if (~isreal(t1(i))) isfirst=0; end;
    else
      t(i,:)=conj(t(i-1,:));
      isfirst=1;
    end
end

% check error
if(nargout>1)
    error=0;
    for i=1:16
      t1=t(i,1); t2=t(i,2); t3=t(i,3);
      error=error+···
      norm([[t1^2 t1 1]*a*[t2^2; t2; 1],[t2^2 t2 1]*b*[t3^2; t3; 1],
    [t3^2 t3 1]*c'*[t1^2; t1; 1]]);
    end
end
```

References

[1] V.E. Gough, S.G. Whitehall, Universal tyre test machine, in: Proceedings of the 9th International Technical Congress, FISITA, 1962, p. 177.

[2] D. Stewart, A platform with six degrees of freedom, Proc. Inst. Mech. Eng., Part I 180 (15) (1966) 371–386.

[3] M. Valenti, Machine tools get smarter, ASME Mech. Eng. 117 (11) (1995) 70–75.

[4] F. Pierrot, P. Dauchez, A. Fournier, Fast parallel robots, J. Robotic Syst. 8 (6) (1991) 829–840.

[5] F. Behi, Kinematic analysis for a six-degree-of-freedom 3-PRPS parallel mechanism, IEEE J. Robotics Autom. 4 (5) (1988) 561–565.

[6] R.I. Alizade, N.R. Tagiyev, J. Duffy, A forward and reverse displacement analysis of a 6-DOF in-parallel manipulator, Mech. Mach. Theory 29 (1) (1994) 115–124.

[7] J.W. Kim et al., Performance analysis of parallel manipulator architectures for CNC machining, in: Proceedings of the 1997 ASME IMECE Symposium on Machine Tools, Dallas, TX, 1997.

[8] G.R. Dunlop, T.P. Jones, Position analysis of a 3-DOF parallel manipulator, Mech. Mach. Theory 32 (8) (1997) 903–920.

[9] J. Nielsen, B. Roth, On the kinematic analysis of robotic mechanisms, Int. J. Robotics Res. 18 (12) (1999) 1147–1160.

[10] P. Nanua, K.J. Waldron, V. Murthy, Direct kinematic solution of a Stewart platform, IEEE Trans. Robotics Autom. 6 (1990) 438–444.

[11] M.L. Husty, An algorithm for solving the direct kinematics of general Stewart–Gough platforms, Mech. Mach. Theory 31 (4) (1996) 365–380.

[12] P. Dietmaier, The Stewart–Gough platform of general geometry can have 40 real postures, in: J. Lenarcic, M. Husty (Eds.), Advances in Robot Kinematics: Analysis and Control, Kluwer Academic Publishers, Dordrecht, 1998, pp. 7–16.