Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea - May 21-26, 2001

Algebraic Elimination-Based Real-Time Forward Kinematics of the 6-6 Stewart Platform with Planar Base and Platform

Tae-Young Lee and Jae-Kyung Shim*
Dept. of Mechanical Engineering, Korea University
1, 5-ka Anam-dong, Sungbuk-ku, Seoul, South Korea 136-701
* phone: +82-2-3290-3362; fax: +82-2-926-9290
e-mail: jkshim@korea.ac.kr

Abstract

This paper presents the closed-form forward kinematics of the 6-6 Stewart platform with planar base and moving platform. Based on algebraic elimination method, it first derives a 20th-degree univariate equation from the determinant of the final Sylvester's matrix of which size is only 4×4. Then, it finds all solutions corresponding to the possible configurations of the platform for a given set of leg lengths. The proposed algorithm requires fairly less computation time enough to be used for real-time applications than the existing ones. Unlike numerical iterative schemes, this algorithm demands no initial estimate and is free from the problem that it fails to converge to the actual solution within limited time. The presented method has been implemented in C language and a numerical example is given to confirm the effectiveness and the accuracy of the developed algorithm.

1 Introduction

The forward kinematics of the Stewart platform is to find the postures of the moving platform for a given set of leg lengths. In general, the forward kinematics of a parallel manipulator is known to be more complicated than the serial one because the formulation of necessary kinematic conditions generates highly nonlinear equations with multiple solutions. A numerical iterative scheme, such as Newton-Raphson method, can be applied to this problem. However, such method not only demands an initial estimate that should be fairly close to the solution of the current configuration, but also cannot guarantee the convergence to the actual solution.

An approach, free from the aforementioned problems, is to find all the solutions, and then select the actual one by the criterion of initial assembly mode or the configuration that the platform had right before the current time. However, when a set of equations is highly nonlinear, finding all the solutions is complicated. One of the well-known methods to provide all solutions is algebraic elimination, which usually changes a set of nonlinear algebraic equations into a univariate equation that can be readily solved by various efficient numerical algorithms.

There have been many literatures on the elimination methods for the forward kinematics of different types and geometry. Innocenti and Parenti-Castelli derived a 16th-degree univariate polynomial on the type 6-3 [1] and a 40th on the type 5-5 [2]. Innocenti [3] found that the forward kinematics of the type 6-4 admits of 32 closure configurations in the complex field, and Nielson and Roth [4] reduced a system of equations for the type 6-5 to a 40th-degree univariate after factoring out trivial roots. For the case of planar base and moving platform, Lin et al. dealt with the type 4-4 [5] and 4-5 [6] and Chen and Song [7] with the type 4-6, while the type 6-6 was studied by Wen and Liang [8] and Zhang and Song [9] independently. For the general 6-6 Stewart platform, Husty [10] produced a 40th-degree univariate equation by finding the greatest common divisor of the intermediate polynomials of degree 320, while Innocenti [11] derived it from the two 56th-degree univariate equations. Dhingra et al. [12] used Gröbner-Sylvester hybrid method to obtain a 40th-degree polynomial directly from the 68×68 Sylvester's matrix formed by calculated Gröbner basis.

All the approaches mentioned above provide algorithms to obtain all solutions for various types of Stewart platforms, but were not intended for real-time applications.

The objective of this paper is to develop an algorithm to provide all the solutions of the forward kinematic analysis of the 6-6 Stewart platform with planar base and moving platform in real-time. The proposed algorithm first derives a 20th-degree univariate polynomial from the determinant of the final 4×4 Sylvester's matrix. Therefore, when implemented in a computer language, it requires so less computation time to find all the solutions than the existing algorithms. It has been implemented in C language using conventional double precision data. A numerical example is given to verify the effectiveness and accuracy of the developed algorithm for real-time computation.

2 Kinematic Constraint Equations

Fig. 1 shows a kinematic model of the 6-6 Stewart platform with planar base and moving platform. All the joints of its base and moving platform are located in respective planes. The six inputs necessary to describe the location and orientation of the upper platform are the leg lengths controlled by each prismatic joint. For convenience sake, the origin of the base frame X-Y-Z is chosen coincident with A_1 and the Z-axis is set to be upward and perpendicular to the base. Similarly, a frame x-y-z is attached to the moving platform with its origin at B_1 and the z-axis is upward and perpendicular to the moving platform. Letting $\mathbf{a}_i$ denote the position vector A_i in X-Y-Z and $\mathbf{b}_i$ denote the position vector B_i in x-y-z, all the z-component of $\mathbf{a}_i$ and $\mathbf{b}_i$ are zeros. The position vector $\mathbf{p}$ indicates the location of the origin of x-y-z with respect to X-Y-Z. With given leg lengths, the kinematic constraint equations corresponding to the conditions of constant length of each leg are as follows

$$(\mathbf{p}+\mathbf{R}\mathbf{b}_i-\mathbf{a}_i)^T(\mathbf{p}+\mathbf{R}\mathbf{b}_i-\mathbf{a}_i)=L_i^2, \qquad i=2,\ldots,6 \quad (1)$$

$$\mathbf{p}^T\mathbf{p}=L_1^2 \quad (2)$$

where L_i is the i-th leg length and $\mathbf{R}$ is a rotation matrix. Using Cayley's formula [13], $\mathbf{R}$ can be expressed as

$$\mathbf{R}=[\mathbf{I}-\mathbf{C}]^{-1}[\mathbf{I}+\mathbf{C}] \quad (3)$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\mathbf{C}$ is an arbitrary 3×3 skew symmetric matrix with three independent parameters, that can be

$$\mathbf{C}=\begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix} \quad (4)$$

Substituting (4) into Eq. (3) yields

$$\mathbf{R}=\Delta^{-1}\begin{bmatrix} 1+c_1^2-c_2^2-c_3^2 & 2(c_1c_2-c_3) & 2(c_3c_1+c_2) \\ 2(c_1c_2+c_3) & 1-c_1^2+c_2^2-c_3^2 & 2(c_2c_3-c_1) \\ 2(c_3c_1-c_2) & 2(c_2c_3+c_1) & 1-c_1^2-c_2^2+c_3^2 \end{bmatrix} \quad (5)$$

where $\Delta=1+c_1^2+c_2^2+c_3^2$. With the relation (2), Eqs. (1) can be expanded as

$$-2\mathbf{a}_i^T\mathbf{p}+2\mathbf{b}_i^T\mathbf{R}^T\mathbf{p}-2\mathbf{a}_i^T\mathbf{R}\mathbf{b}_i+\mathbf{a}_i^T\mathbf{a}_i+\mathbf{b}_i^T\mathbf{b}_i-L_i^2+L_1^2=0, \qquad i=2,\ldots,6 \quad (6)$$

Assembling the rotational parameters into the vector $\mathbf{c}=[c_1, c_2, c_3]^T$, the six equations (2) and (6) contain the unknown translation vector $\mathbf{p}$ and the vector $\mathbf{c}$. These variables should be computed to determine the postures of the moving platform with the six leg lengths.

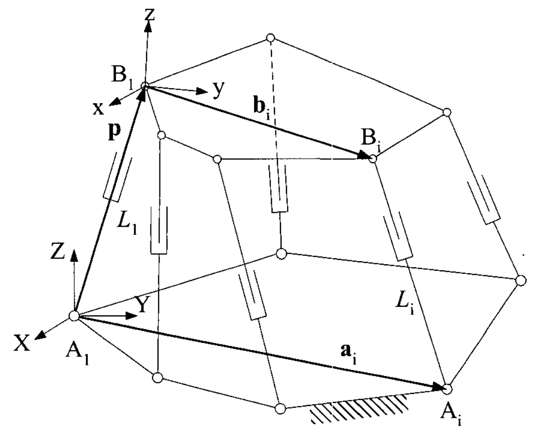

Figure 1. The 6-6 Stewart platform.

3 Solution Procedure

3.1 Modification of the constraint equations.

For more efficient solution procedure, another translation vector $\mathbf{q}$ can be introduced as follows [11, 14]

$$\mathbf{q}=\mathbf{R}^T\mathbf{p} \quad (7)$$

Accordingly Eq. (6) can be rewritten as follows after divided by two

$$-\mathbf{a}_i^T\mathbf{p}+\mathbf{b}_i^T\mathbf{q}-\mathbf{a}_i^T\mathbf{R}\mathbf{b}_i+\frac{1}{2}\left(\mathbf{a}_i^T\mathbf{a}_i+\mathbf{b}_i^T\mathbf{b}_i-L_i^2+L_1^2\right)=0, \qquad i=2,\ldots,6 \quad (8)$$

By the relations (3) and (7), the following equation should be satisfied.

$$[\mathbf{I}-\mathbf{C}]\mathbf{p}-[\mathbf{I}+\mathbf{C}]\mathbf{q}=\mathbf{0} \quad (9)$$

If we let $\mathbf{p}=[\,p_x, p_y, p_z\,]^T$ and $\mathbf{q}=[\,q_x, q_y, q_z\,]^T$, Eq. (9) can be written in the following scalar equations

$$p_x+c_3p_y-c_2p_z-q_x+c_3q_y-c_2q_z=0 \quad (10)$$

$$-c_3p_x+p_y+c_1p_z-c_3q_x-q_y+c_1q_z=0 \quad (11)$$

$$c_2p_x-c_1p_y+p_z+c_2q_x-c_1q_y-q_z=0 \quad (12)$$

Now the unknown vectors are $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{c}$ that should be determined from nine scalar equations (2), (8), and (10)-(12).

3.2 Intermediate polynomials in c_1, c_1, and c_3 only.

In matrix form, Eq. (8) can be arranged as follows

$$\mathbf{Mu}=\begin{bmatrix} -a_{2x} & -a_{2y} & b_{2x} & b_{2y} & F_2 \\ -a_{3x} & -a_{3y} & b_{3x} & b_{3y} & F_3 \\ -a_{4x} & -a_{4y} & b_{4x} & b_{4y} & F_4 \\ -a_{5x} & -a_{5y} & b_{5x} & b_{5y} & F_5 \\ -a_{6x} & -a_{6y} & b_{6x} & b_{6y} & F_6 \end{bmatrix}\begin{bmatrix} p_x \\ p_y \\ q_x \\ q_y \\ 1 \end{bmatrix}=\mathbf{0} \quad (13)$$

where

$$F_i=-\mathbf{a}_i^T\mathbf{R}\mathbf{b}_i+\frac{1}{2}\left(\mathbf{a}_i^T\mathbf{a}_i+\mathbf{b}_i^T\mathbf{b}_i-L_i^{\,2}+L_1^{\,2}\right),\quad i=2,\ldots,6 \quad (14)$$

and $\mathbf{a}_i=[a_{ix}, a_{iy}, 0]^T$ and $\mathbf{b}_i=[b_{ix}, b_{iy}, 0]^T$ (i=2,…,6). The linear system (13) is over-constrained, so there should be linear dependency for the system to have a solution. That is

$$\det(\mathbf{M})=0 \quad (15)$$

The condition (15) leads to the following polynomial in c_1, c_2, and c_3

$$\Phi_1(c_1,c_2,c_3)\equiv g_1+g_2\,c_3+g_3\,c_3^2+g_4\,c_1^2+g_5\,c_1c_2+g_6\,c_2^2=0 \quad (16)$$

where, g_i(i=1,…,6) are real coefficients determined by input data only.

Adding $c_1\times$Eq.(10) to $c_2\times$Eq.(11) yields

$$(c_1-c_2c_3)p_x+(c_2+c_1c_3)p_y-(c_1+c_2c_3)q_x-(c_2-c_1c_3)q_y=0 \quad (17)$$

If Eq. (17) replaces the last row of the matrix $\mathbf{M}$ in (13), another linear system is obtained as follows

$$\mathbf{M'u}=\begin{bmatrix} -a_{2x} & -a_{2y} & b_{2x} & b_{2y} & F_2 \\ -a_{3x} & -a_{3y} & b_{3x} & b_{3y} & F_3 \\ -a_{4x} & -a_{4y} & b_{4x} & b_{4y} & F_4 \\ -a_{5x} & -a_{5y} & b_{5x} & b_{5y} & F_5 \\ c_1-c_2c_3 & c_2+c_1c_3 & -c_1-c_2c_3 & -c_2+c_1c_3 & 0 \end{bmatrix}\begin{bmatrix} p_x \\ p_y \\ q_x \\ q_y \\ 1 \end{bmatrix}=\mathbf{0} \quad (18)$$

Similarly, the system (18) can have a solution if and only if

$$\det(\mathbf{M'})=0 \quad (19)$$

The condition (19) leads to another polynomial in c_1, c_2, and c_3 as follows

$$\Phi_2(c_1,c_2,c_3)\equiv T_1\,c_1+T_2\,c_2+\sum_{j=0}^{3}T_{3j}\,c_1^{3-j}c_2^{j}=0 \quad (20)$$

where

$$T_i(c_3)=\sum_{k=0}^{3}t_{i,k}c_3^k,\qquad i=1,2$$
$$T_{3j}(c_3)=\sum_{k=0}^{1}t_{3j,k}c_3^k,\quad j=0,\ldots,3 \quad (21)$$

and t's with subscripts are real coefficients determined by input data only.

Up to this point, we have two equations (16) and (20) in three unknowns, c_1, c_2 and c_3. Hence, we need one or more equations that reflect equation (2), which can be expanded as

$$p_x^2+p_y^2+p_z^2-L_1^2=0 \quad (22)$$

If we solve the following minor system of Eq. (13) symbolically, the solution $[\overline{p}_x,\overline{p}_y,\overline{q}_x,\overline{q}_y]$ for $[p_x,p_y,q_x,q_y]$ are determined in terms of c_1, c_2, and c_3.

$$\begin{bmatrix} -a_{2x} & -a_{2y} & b_{2x} & b_{2y} & F_2 \\ -a_{3x} & -a_{3y} & b_{3x} & b_{3y} & F_3 \\ -a_{4x} & -a_{4y} & b_{4x} & b_{4y} & F_4 \\ -a_{5x} & -a_{5y} & b_{5x} & b_{5y} & F_5 \end{bmatrix}\begin{bmatrix} p_x \\ p_y \\ q_x \\ q_y \\ 1 \end{bmatrix}=\mathbf{0} \quad (23)$$

Letting $\mathbf{v}[v_1,v_2,v_3,v_4]=[\overline{p}_x,\overline{p}_y,\overline{q}_x,\overline{q}_y]$ for the simplicity of representation, the solution of the system (23) can be written in the following forms

$$v_i=\frac{s_{i1}+s_{i2}\,c_3+s_{i3}\,c_3^2+s_{i4}\,c_1^2+s_{i5}\,c_1c_2+s_{i6}\,c_2^2}{1+c_1^2+c_2^2+c_3^2},\quad i=1,\ldots,4 \quad (24)$$

If the symbolic solution for $\mathbf{v}$ is substituted into Eqs. (10) and (11), we can find the symbolic solution $[\overline{p}_{z1},\overline{q}_{z1}]$ for $[p_z,q_z]$. In the same way, another solution $[\overline{p}_{z2},\overline{q}_{z2}]$ can be obtained by Eqs. (10) and (12). The two solutions $\overline{p}_{z1}$ and $\overline{p}_{z2}$ for p_z can be expressed as follows

$$p_{z1}=\frac{N_1(c_1,c_2,c_3)}{D_1(c_1,c_2,c_3)}=\frac{N_{1,1}+\sum_{j=0}^{2}N_{1,2j}c_1^{2-j}c_2^j+\sum_{j=0}^{4}n_{1,3j}c_1^{4-j}c_2^j}{c_1(1+c_1^2+c_2^2+c_3^2)} \quad (25)$$

$$p_{z2}=\frac{N_2(c_1,c_2,c_3)}{D_2(c_1,c_2,c_3)}=\frac{N_{2,1}+\sum_{j=0}^{2}N_{2,2j}c_1^{2-j}c_2^j+\sum_{j=0}^{4}n_{2,3j}c_1^{4-j}c_2^j}{c_2(1+c_1^2+c_2^2+c_3^2)} \quad (26)$$

where

$$N_{i,1}(c_3)=\sum_{k=0}^{3} n_{i,1k}c_3^k, \qquad i=1,2$$
$$N_{i,2j}(c_3)=\sum_{k=0}^{2} n_{i,2jk}c_3^k, \quad j=0,1,2 \tag{27}$$

and n's with subscripts are constants depending on input data only. By the following, the third solution p_{z3} can be generated

$$p_{z3}=\frac{N_1(c_1,c_2,c_3)-N_2(c_1,c_2,c_3)}{D_1(c_1,c_2,c_3)-D_2(c_1,c_2,c_3)} \tag{28}$$

Hence, there are three symbolic solutions for **p**, i.e. $[\bar{p}_x,\bar{p}_y,\bar{p}_{z1}]$, $[\bar{p}_x,\bar{p}_y,\bar{p}_{z2}]$, and $[\bar{p}_x,\bar{p}_y,\bar{p}_{z3}]$. If we substitute the three solutions into Eq. (22) one by one, three polynomials of the following form are produced after rationalization

$$\Phi_{i+2}(c_1,c_2,c_3)=U_{i,1}+\sum_{j=0}^{2}U_{i,2j}c_1^{2-j}c_2^j+\sum_{j=0}^{4}U_{i,3j}c_1^{4-j}c_2^j+$$
$$\sum_{j=0}^{6}U_{i,4j}c_1^{6-j}c_2^j+\sum_{j=0}^{8}u_{i,5j}c_1^{8-j}c_2^j, \quad i=1,2,3 \tag{29}$$

where

$$U_{i,1}(c_3)=\sum_{k=0}^{6}u_{i,1k}c_3^k, \qquad U_{i,2j}(c_3)=\sum_{k=0}^{5}u_{i,2jk}c_3^k,$$
$$U_{i,3j}(c_3)=\sum_{k=0}^{4}u_{i,3jk}c_3^k, \quad U_{i,4j}(c_3)=\sum_{k=0}^{2}u_{i,4jk}c_3^k \tag{30}$$

and u's with subscripts are real coefficients.

Now, the five intermediate polynomials Φ_i(i=1,…,5) in (16), (20), and (29) have been derived.

3.3 Modification of the polynomials Φ_i (i=3,4,5)

In order to derive a univariate equation in c_3, the two unknowns c_1 and c_2 should be eliminated from the equations Φ_i (i=1,…,5). However, since the five equations contain so many power products of c_1 and c_2 that too large-sized Sylvester's matrix is expected. In this section, a method to reduce the number of power products is presented. It eventually reduces the size of the final Sylvester's matrix to 4×4, so the required computation time can be greatly decreased.

Writing $\Phi_1\times c_1^2$, $\Phi_1\times c_1c_2$, $\Phi_1\times c_2^2$, $\Phi_2\times c_1$, and $\Phi_1\times c_2$ in matrix form, we have

$$\begin{bmatrix} g_4 & g_5 & g_6 & 0 & 0 \\ 0 & g_4 & g_5 & g_6 & 0 \\ 0 & 0 & g_4 & g_5 & g_6 \\ T_{30} & T_{31} & T_{32} & T_{33} & 0 \\ 0 & T_{30} & T_{31} & T_{32} & T_{33} \end{bmatrix}\begin{bmatrix} c_1^4 \\ c_1^3c_2 \\ c_1^2c_2^2 \\ c_1c_2^3 \\ c_2^4 \end{bmatrix}=-\begin{bmatrix} (g_1+g_2c_3+g_3c_3^2)c_1^2 \\ (g_1+g_2c_3+g_3c_3^2)c_1c_2 \\ (g_1+g_2c_3+g_3c_3^2)c_2^2 \\ T_1c_1^2+T_2c_1c_2 \\ T_1c_1c_2+T_2c_2^2 \end{bmatrix} \tag{31}$$

Solving the system (31) symbolically regarding all power products $c_1^{4-j}c_2^j(j=0,...,4)$ as linear unknowns, we obtain the following expression

$$c_1^{4-j}c_2^j=\frac{H_{j1}c_1^2+H_{j2}c_1c_2+H_{j3}c_2^2}{E}, \quad j=0,...,4 \tag{32}$$

where

$$H_{jk}(c_3)=\sum_{m=0}^{4}h_{jkm}c_3^m, \quad j=0,...,4,\ k=1,2,3$$
$$E(c_3)=d_0+d_1c_3+d_2c_3^2 \tag{33}$$

and h_{jkm} and d_i(i=0,1,2) are real constants depending on input data only.

Using Eq. (32), we can reduce the degrees of Eqs. (29) with respect to c_1 and c_2. The terms $c_1^{4-j}c_2^j$ (j=0,…,4), $c_1^{6-j}c_2^j$ (j=0,…,6), and $c_1^{8-j}c_2^j$ (j=0,…,8) in Eq. (29) can be changed into $c_1^{2-j}c_2^j$ (j=0,1,2) by successive substitutions using (32). As a result, the equations Φ_i (i=3,4,5) in (29) are transformed into the following form

$$\Phi'_{i+2}(c_1,c_2,c_3)=\frac{W_{i1}+W_{i2}c_1^2+W_{i3}c_1c_2+W_{i4}c_2^2}{E^3}, \ i=1,2,3 \tag{34}$$

where

$$W_{i1}(c_3)=\sum_{k=0}^{12}w_{i1k}c_3^k, \qquad i=1,2,3$$
$$W_{ij}(c_3)=\sum_{k=0}^{10}w_{ijk}c_3^k, \quad i=1,2,3,\ j=2,3,4 \tag{35}$$

and w's with subscripts are real coefficients. In Eqs. (34), there remain only four power products of c_1 and c_2, which are $c_1^{2-j}c_2^j$ (j=0,1,2) and $1(=c_1^0c_2^0)$.

3.4 Deriving a univariate polynomial in c_3

Combining Φ_1 of Eq. (16) with Φ'_i (i=3,4,5) of Eq. (34) leads to the following linear system

$$\mathbf{S}\hat{\mathbf{c}}=\begin{bmatrix} G_1 & g_4 & g_5 & g_6 \\ \frac{W_{11}}{E^3} & \frac{W_{12}}{E^3} & \frac{W_{13}}{E^3} & \frac{W_{14}}{E^3} \\ \frac{W_{21}}{E^3} & \frac{W_{22}}{E^3} & \frac{W_{23}}{E^3} & \frac{W_{24}}{E^3} \\ \frac{W_{31}}{E^3} & \frac{W_{32}}{E^3} & \frac{W_{33}}{E^3} & \frac{W_{34}}{E^3} \end{bmatrix}\begin{bmatrix} 1 \\ c_1^2 \\ c_1c_2 \\ c_1^2 \end{bmatrix}=\mathbf{0} \tag{36}$$

where $G_1=g_1+g_2c_3+g_3c_3^2$. E and W_{ij} are given in Eqs. (33) and (35). The over-constrained system (36) has a solution, if and only if

$$\det(\mathbf{S})=0 \tag{37}$$

The condition (37) yields the following univariate equation after factorization

$$\frac{1}{E^3}\sum_{i=0}^{20} s_i c_3^i = 0 \qquad (38)$$

where s_i (i=0,...,20) are real constants depending on input data only. The numerator of Eq. (38) is a 20th-degree polynomial in c_3. Solving Eq. (38) gives 20 roots c_{3i} (i=1,...,20) for c_3 in the complex domain.

3.5 Back substitution for other unknowns

Linear solving the system, which is obtained by removing any one row from the matrix **S** in Eq. (36) with c_3 replaced by c_{3i} (i=1,...,20), provides the values $(c_1^2)_i$, $(c_1 c_2)_i$, and $(c_2^2)_i$ for c_1^2, $c_1 c_2$, and c_2^2. The values c_{1i} and c_{2i} for c_1 and c_2 can be easily computed in the complex domain as follows

$$\begin{cases} \text{if } (c_1^2)_i \neq 0: \quad c_{1i} = \pm\sqrt{(c_1^2)_i}, \quad c_{2i} = \dfrac{(c_1 c_2)_i}{c_{1i}} \\ \text{if } (c_1^2)_i = 0: \quad c_{1i} = 0, \quad c_{2i} = \pm\sqrt{(c_2^2)_i} \end{cases} \qquad (39)$$

There exist two sets of solutions for each c_{3i}, i.e. if $[c_{1i}, c_{2i}]$ is a solution corresponding to c_{3i}, $[-c_{1i}, -c_{2i}]$ is also a solution. Therefore, the moving platform can have 40 solutions for a given set of leg lengths, though the degree of the univariate equation is only 20. The values of p_{xi} and p_{yi} for p_x and p_y of the translational parameter **p** are computed from Eq. (24), and p_{zi} for p_z is determined by

$$\begin{cases} \text{if } c_{1i} \neq 0: & \text{by Eq. (25)} \\ \text{if } c_{1i} = 0, c_{2i} \neq 0: & \text{by Eq. (26)} \\ \text{if } c_{1i}, c_{2i} = 0: & p_{zi} = \pm\sqrt{L_1^2 - p_{xi}^2 - p_{yi}^2} \end{cases} \qquad (40)$$

Now 40 sets of solutions of the necessary parameters to describe the posture of the moving platform have been determined.

4 Computer Implementation

One of the ways to get the 20th-degree univariate equation is to compute the determinant of the Sylvester's matrix symbolically. But it is not only difficult to implement but also inadequate for real-time implementation. Instead, 21 coefficients of the 20th-degree polynomial can be computed by the numerical expansion of the polynomial coefficients. If we substitute 21 arbitrarily-selected values $\bar{c}_{3i}$ (i = 1,...,21) for c_3, the following linear system is obtained

$$\begin{bmatrix} 1 & \bar{c}_{31} & \bar{c}_{31}^2 & \cdots & \bar{c}_{31}^{19} & \bar{c}_{31}^{20} \\ 1 & \bar{c}_{32} & \bar{c}_{32}^2 & \cdots & \bar{c}_{32}^{19} & \bar{c}_{32}^{20} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \bar{c}_{320} & \bar{c}_{320}^2 & \cdots & \bar{c}_{320}^{19} & \bar{c}_{320}^{20} \\ 1 & \bar{c}_{321} & \bar{c}_{321}^2 & \cdots & \bar{c}_{321}^{19} & \bar{c}_{321}^{20} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{19} \\ s_{20} \end{bmatrix} = \begin{bmatrix} E^3(\bar{c}_{31})\det(\mathbf{S}|_{c_3=\bar{c}_{31}}) \\ E^3(\bar{c}_{32})\det(\mathbf{S}|_{c_3=\bar{c}_{32}}) \\ \vdots \\ E^3(\bar{c}_{320})\det(\mathbf{S}|_{c_3=\bar{c}_{320}}) \\ E^3(\bar{c}_{321})\det(\mathbf{S}|_{c_3=\bar{c}_{321}}) \end{bmatrix} \qquad (41)$$

Solving (41) provides the values of all the coefficients, in which we can save computation time by using a predetermined inverse of the coefficient matrix. In this manner, the symbolic computation of determinants and factorization is replaced with the calculation of numeric matrices.

For solving a univariate polynomial, many efficient numerical algorithms are available. In this paper, a function in Cephes Math library [15] is used, which shows fast and exact performance.

Because the proposed algorithm is not sensitive to round-off errors, it can be readily implemented in C language using the double precision data type of 15 significant digits.

5 Numerical Example

A numerical example is presented to demonstrate the validity of the proposed algorithm for a 6-6 Stewart platform. The geometric parameters of the platform are given as

$\mathbf{a}_1=[0, 0, 0]^T$, $\mathbf{a}_2=[62, 0, 0]^T$, $\mathbf{a}_3=[62, 11, 0]^T$,
$\mathbf{a}_4=[42, 38, 0]^T$, $\mathbf{a}_5=[32, 39, 0]^T$, $\mathbf{a}_6=[7, 13, 0]^T$,
$\mathbf{b}_1=[0, 0, 0]^T$, $\mathbf{b}_2=[14, 0, 0]^T$, $\mathbf{b}_3=[47, 13, 0]^T$,
$\mathbf{b}_4=[46, 27, 0]^T$, $\mathbf{b}_5=[23, 45, 0]^T$, $\mathbf{b}_6=[16, 42, 0]^T$.

To check the accuracy of the implemented algorithm, leg lengths of inputs are computed from the following predetermined configuration

$$\mathbf{p} = [\,12, 23, 96\,]^T, \qquad \mathbf{c} = [\,1.0, -1.2, 0.8\,]^T.$$

The exact values of leg lengths are

$L_1 = 99.4434512675420$, $L_2 = 122.382476638755$,

$L_3 = 156.014956547975$, $L_4 = 153.949953670971$,

$L_5 = 136.270060584725$, $L_6 = 117.805089939638$.

Using $\mathbf{a}_i$, $\mathbf{b}_i$, and L_i (i=1,...,6) as inputs, the 40 sets of solutions shown in Table 1 are obtained. In Table 1, one result written in boldface is almost coincident with given **p** and **c**. The exact values obtained by the implemented algorithm in this case are

$$\mathbf{p} = [11.9999999975, 23.0000000219, 96.0000000247]^T,$$

$$\mathbf{c} = [1.00000000060, -1.20000000113, 0.80000000018]^T.$$

The total computation time to determine all the configurations by the developed algorithm is within 1.0 msec on a PC (PentiumII-350MHz), and almost uniform computation time is maintained over various examples.

Table 1. Solutions for numerical example

Sol. No.	$\pm c_1$	$\pm c_2$	c_3	p_x	p_y	$\pm p_z$
1, 2	0.88339*i*	0.47909*i*	0.22003	-205.561	129.287	221.544*i*
3, 4	0.86303*i*	0.51653*i*	0.22690	-196.273	289.611	335.422*i*
5, 6	0.86575*i*	-1.06856	0.46212	1.95795	209.645	184.569*i*
7, 8	0.09363*i*	-1.16458	0.55470	-71.9190	-280.446	-271.916*i*
9, 10	0.22625*i*	-1.16619	0.56982	-59.1385	90.3190	-42.1681*i*
11, 12	0.43373*i*	-0.95972	0.57510	41.3751	-266.270	-250.450*i*
13, 14	1.98573*i*	0.30557*i*	-1.68276	101.669	9.61672	23.2413*i*
15, 16	**1.00000**	**-1.20000**	**0.80000**	**12.0000**	**23.0000**	**96.0000**
17, 18	1.07477*i*	3.22158*i*	-3.15089	-19.0555	-2056.30	-2053.99*i*
19, 20	1.09654*i*	3.15672*i*	-3.19373	619.574	37565.4	37570.4*i*
21, 22	0.55383	-0.82548	0.66527	12.5852	-0.05356	98.6439
23, 24	2.01113*i*	0.28091*i*	-1.77265	-893.766	-256.840	-924.605*i*
25, 26	2.45917*i*	-4.08233	-4.69370	609.758	-926.546	1104.53*i*
27, 28	2.45726*i*	-4.24973	-4.70415	-189.485	283.613	-326.176*i*
29-32	0.00327 ∓ 0.98280*i*	0.036718 ± 0.32262*i*	0.314984 ∓ 0.00850*i*	-51.5323 ± 10.3008*i*	-3300.01 ± 1408.16*i*	1408.70 ± 3299.13*i*
33-36	0.39527 ∓ 3.90366*i*	0.468082 ± 0.25963*i*	-2.23281 ∓ 5.07359*i*	18.8430 ± 0.3434*i*	19.3024 ± 6.3306*i*	95.9341 ∓ 1.3412*i*
37-40	-10.4886 ∓ 8.26754*i*	-4.8979 ± 37.950*i*	8.96245 ∓ 8.11223*i*	9.29012 ± 2.87628*i*	-30.9662 ∓ 48.1556*i*	-106.647 ± 14.2332*i*

6 Conclusions

This paper presents an algebraic elimination-based method for the real-time forward kinematics of the 6-6 Stewart platform with planar base and moving platform without extra sensors. By using the presented algorithm, the computation time is greatly reduced to the extent that it can be efficiently used for real-time applications. The effectiveness and exactness of the algorithm are verified by a numerical example.

7 References

[1] Innocenti, C., and Parenti-Castelli, V., 1990, "Direct Position Analysis of the Stewart Platform Mechanism," *Mechanism and Machine Theory.* Vol. 25, No. 6, pp. 611-621.

[2] Innocenti, C., and Parenti-Castelli, V., 1993, "Closed-Form Direct Position Analysis of a 5-5 Parallel Mechanism," *ASME Journal of Mechanical Design*, Vol. 115, pp. 515-522.

[3] Innocenti, C., 1995, "Direct Kinematics in Analytical Form of the 6-4 Fully-Parallel Mechanism," *ASME Journal of Mechanical Design*, Vol. 117, pp. 89-95.

[4] Nielson, J., and Roth, B., 1996, "The Direct Kinematics of the General 6-5 Stewart-Gough Mechanism," *Recent Advances in Robot Kinematics*, Kluwer Academic Publishers, pp. 7-16.

[5] Lin, W., Griffis, M., and Duffy, J., 1992 "Forward Displacement Analyses of the 4-4 Stewart Platforms," *ASME Journal of Mechanical Design*, Vol. 114, pp. 444-450.

[6] Lin, W., Crane, C., and Duffy, J., 1994 "Closed-Form Forward Analysis of the 4-5 In-Parallel Platforms," *ASME Journal of Mechanical Design*, Vol. 116, pp. 47-53.

[7] Chen, N., and Song, S., 1994 "Direct Position Analysis of the 4-6 Stewart Platforms," *ASME Journal of Mechanical Design*, Vol. 116, pp. 61-66.

[8] Wen, F., and Liang, C., 1994 "Displacement Analysis of the 6-6 Stewart Platform Mechanisms," *Mechanism and Machine Theory*, Vol. 29, No. 4, pp.547-557.

[9] Zhang, C., and Song, S., 1994 "Forward Position Analysis of Nearly General Stewart Platforms," *ASME Journal of Mechanical Design*, Vol. 116, pp. 54-60.

[10] Husty, M.L., 1996, "An Algorithm for Solving the Direct Kinematics of General Stewart-Gough Platforms," *Mechanism and Machine Theory*, Vol. 31, No. 4, pp. 365-380.

[11] Innocenti, C., 1998, "Forward Kinematics in Polynomial Form of the General Stewart Platform," *25th Biennial Mechanisms Conference,* Atlanta GA, Paper: DETC98/MECH-5894.

[12] Dhingra, A., Almadi, A., and Kohli, D., 1998, "A Gröbner-Sylvester Hybrid Method for Closed-Form Displacement Analysis of Mechanisms," *25th Biennial Mechanisms Conference, Atlanta GA,* Paper: DETC98/MECH-5969.

[13] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, North-Holland Publishing Company, New York, pp. 9-11.

[14] Lazard, D., 1993, "On the Representation of Rigid-Body Motions and Its Application to Generalized Platform Manipulator," *Computational Kinematics*, Kluwer Academic Publishers, pp. 175-181.

[15] Moshier, S. L., *Cephes Math Library.* Available from http:// people.ne.mediaone.net/moshier/index.html.