Fast Numerical Solution to Forward Kinematics of General Stewart Mechanism Using Quaternion

Yang Xiaolong(杨小龙)[1], *Wu Hongtao*(吴洪涛)[1*], *Chen Bai*(陈柏)[1], *Zhu Liucun*(朱留存)[2], *Xun Junxiang*(荀俊祥)[2]

1. College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P. R. China;
2. Changzhou Vraic Automatic Equipment Technology Co., Ltd., Changzhou, 213104, P. R. China

(Received 28 October 2013; revised 27 January 2014; accepted 30 January 2014)

Abstract: The forward kinematics of the general Stewart mechanism is studied and a fast numerical method is presented. Quaternion is utilized to model the forward kinematics and the equations are merely a system of quadratic ones. The numerical method is a nice simplification of the Newton-Raphson method when applied to this system. A simulation of the movement control of the Stewart mechanism is accomplished, confirming the effectiveness of the proposed algorithm in real-time conditions.

Key words: forward kinematics; Stewart mechanism; quaternion

CLC number: TP242 **Document code**: A **Article ID**: 1005-1120(2014)04-0377-09

1 Introduction

Stewart mechanism (also named as Stewart platform) is a kind of parallel kinematic structures, consisting of one upper platform (mobile platform), a lower one (base) and six parallel rods. Each rod is able to stretch out and draw back freely. These two platforms are connected via six such rods with spherical joints at the higher positions and universal joints at the lower positions. The base is stationary and fixed to the ground. The rods have an in-built mechanism that allows changing the length of each individual one. The desired position and orientation of the mobile platform are achieved by changing the lengths of the six rods independently, that is to say, the six transitional degrees of freedom are transformed into three positional and three orientational ones. Compared with the serial mechanism, it has some inherent advantages. In detail, it possesses a greater stiffness-to-mass ratio, higher base frequencies, which enable it to withstand a relatively large load. In addition, it possesses stronger dynamic performance and stability and higher precision of movement, making it competent for precise tasks. Since it was proposed in 1965, its kinematics, singularity, workspace and dexterity, dynamics, control and structural design have been researched in depth and extensively. Nowadays large quantities of the mechanism have been widely used in systems of motion simulation, micro-displacement positioning devices, visualizing haptic devices, industrial and medical robots, telescopes and other aspects.

Although many advantages of the Stewart mechanism make it an ideal solution to some special applications such as the machining center and radio telescopes, it is real difficult to find the stable solution of the kinematics with a low calculation error and time consuming due to its high degree of coupling. The inverse kinematics problem for the Stewart platform is defined as to find a series of rod lengths according to the desired pose

Foundation items: Supported by the National Natural Science Foundation of China (51375230); the Jiangsu Provincial Science and Technology Support Program (BE2012052).

* **Corresponding author**: Wu Hongtao, Professor, E-mail: mehtwu@126.com.

(position and orientation) of the moving platform. The solution to this problem is indeed not at all complex and can be computed in a very short time because the mathematical expression of every rod length is independent to each other. Furthermore, the computation of length for each rod can be carried out independently in parallel, which can additionally speed up the process. On the other hand, there is no closed-form solution for the forward kinematics problem of the general Stewart mechanism[1], which is defined as to find the position and orientation of the mobile platform while the rod lengths are known. Moreover, the fast solution to forward kinematics plays a most important and significant role on the feedback control and analysis of singularity and workspace. Thus, a fast numerical solution to the forward kinematics problem is a challenging puzzle in the research field of the parallel mechanisms.

There are two categories of methods to solve the forward kinematics problem: Analytical methods and numerical methods. In the aspect of analytical methods, a number of researchers use algebraic formulations to generate a high degree of polynomial or a set of nonlinear equations and focus on finding all the possible roots to the equations by means of algebraic elimination, continuation, interval analysis and so on[2-5]. They have made some progress and those solutions are called assembly modes of the Stewart mechanism. Unfortunately, the variables representing the pose of the mobile platform have not been expressed in explicit forms so far[6-9]. Moreover, finding all possible solutions is not completely solving the forward kinematics problem. We still need some schemes to determine a unique actual pose among all the possible solutions, which is required for practical applications. In some cases, use of auxiliary sensors is one commonly adopted scheme to further lead to a unique solution[10-11]. But such an approach is limited in practical applications due to the expensive price and measurement errors. Then in the aspect of numerical methods, there exists a widely used method named as Newton-Raphson method. Nonlinear algebraic equations can be linearized and transformed into linear equations via this method. If the iterative initial value that we choose is located in the domain of convergence, we can obtain the exact solution[12-14]. Some scholars use neural network algorithm to obtain the initial guess required for Newton-Raphson algorithm to ensure the stability of it[15]. The unique solution can also be obtained by making use of the optimization algorithm such as genetic algorithm and neural network algorithm[16-21]. However, both the genetic algorithm and neural network algorithm take much more time so that they are not very suitable for real-time applications.

The complexity of the forward kinematics problem depends widely on the configuration, the geometrical size and the sensor layout of the Stewart mechanism. Although some promising results have been achieved for certain simplified configuration such as usage of composite spherical joints and parallel layout of joints[8], the pursuit of practical algorithms for the general Stewart mechanism has universal significance and should be continued. Moreover, it is difficult to enable the Stewart platform meet the needs of high-speed and real-time engineering applications just by several exiting numerical algorithms. The paper is a contribution of the problem since the goal is to study the general Stewart mechanism and presents an improved Newton-Raphson method that can satisfy the requirement of real-time applications.

2 Forward Kinematics Based On Quaternion

The forward kinematic equations can be expressed as a variety of mathematically equivalent forms. Every representation of the equations has its advantages and disadvantages which become emphasized when a kind of numerical algorithm is applied. By taking quaternions instead of Euler angles, direction cosine matrices or spinors as rotational coordinates to describe the pose, the forward kinematic equations are mere a set of quadratic polynomial ones.

2.1 Representation of rotation using unit quaternion

In the three-dimensional vector space, the orientation of the fixed-point rotation of a rigid body can be expressed by $\boldsymbol{R}\in \boldsymbol{SO}(3)$. $\boldsymbol{SO}(3)$ is a group that meets the rule of matrix multiplication and defined as $\boldsymbol{SO}(3)=\{\boldsymbol{R}\in\mathbb{R}^{3\times3}:\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}=\boldsymbol{I},\det\boldsymbol{R}=+1\}$[22]. $\boldsymbol{R}$ is a linear operator determined by the rotation axis which is defined by a unit vector $\boldsymbol{n}$ and its amplitude ω. When it is necessary to specify these elements, the rotation is denoted as the operator $\boldsymbol{R}(\omega,\boldsymbol{n})$.

As shown in Fig. 1, any vector $\boldsymbol{x}$ can be decomposed into a sum of two mutually perpendicular parts. One is parallel to the vector $\boldsymbol{n}$ and the other is perpendicular to $\boldsymbol{n}$

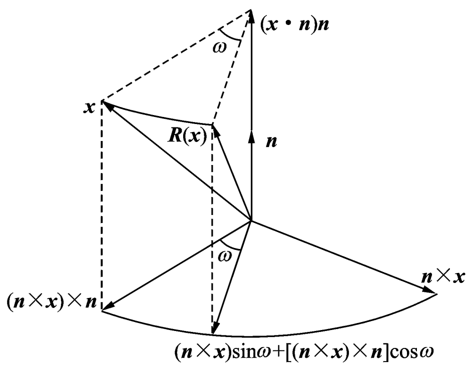
$$\boldsymbol{x}=(\boldsymbol{x}\cdot\boldsymbol{n})\boldsymbol{n}+(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n} \tag{1}$$

Due to the vector $\boldsymbol{n}$, $\boldsymbol{n}\times\boldsymbol{x}$ and $(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n}$ are orthogonal to each other, after a rotation the vector $\boldsymbol{x}$ will be

$$\boldsymbol{R}(\boldsymbol{x})=(\boldsymbol{x}\cdot\boldsymbol{n})\boldsymbol{n}+\boldsymbol{R}[(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n}]=(\boldsymbol{x}\cdot\boldsymbol{n})\boldsymbol{n}+(\boldsymbol{n}\times\boldsymbol{x})\sin\omega+[(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n}]\cos\omega \tag{2}$$

By using the radial and transversal decomposition Eq. (1), Eq. (2) can be written as the well-known Euler-Rodrigues formula

$$\boldsymbol{R}(\boldsymbol{x})=\boldsymbol{x}+(\boldsymbol{n}\times\boldsymbol{x})\sin\omega+[\boldsymbol{n}\times(\boldsymbol{n}\times\boldsymbol{x})](1-\cos\omega) \tag{3}$$



$(\boldsymbol{x}\cdot\boldsymbol{n})\boldsymbol{n}$
ω
$\boldsymbol{x}$
$\boldsymbol{R}(\boldsymbol{x})$
$\boldsymbol{n}$
ω
$\boldsymbol{n}\times\boldsymbol{x}$
$(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n}$
$(\boldsymbol{n}\times\boldsymbol{x})\sin\omega+[(\boldsymbol{n}\times\boldsymbol{x})\times\boldsymbol{n}]\cos\omega$

Fig. 1 Rotation of vector

Now let us consider the set $\mathbb{R}^3\times\mathbb{R}$, whose element is a pair made of a scalar q_0 and a vector $\boldsymbol{q}$. It is expressed as $q=(\boldsymbol{q},q_0)=(q_1\ q_2\ q_3\ q_0)$ or $q=q_1\boldsymbol{i}+q_2\boldsymbol{j}+q_3\boldsymbol{k}+q_0$, with the special rule that $\boldsymbol{i}^2=\boldsymbol{j}^2=\boldsymbol{k}^2=-1$, $\boldsymbol{ij}=-\boldsymbol{ji}=\boldsymbol{k}$, $\boldsymbol{jk}=-\boldsymbol{kj}=\boldsymbol{i}$ and $\boldsymbol{ki}=-\boldsymbol{ik}=\boldsymbol{j}$. Therefore, the law of composition

$$(q,p)\rightarrow qp=\{q_0\boldsymbol{p}+p_0\boldsymbol{q}+\boldsymbol{q}\times\boldsymbol{p},q_0p_0-\boldsymbol{q}\cdot\boldsymbol{p}\} \tag{4}$$

is bilinear of q and p. It is easy to verify that the composition operation is associative but not commutative since it involves a cross product. All of these characteristics make $\mathbb{R}^3\times\mathbb{R}$ an associative algebra. The set $\mathbb{R}^3\times\mathbb{R}$ endowed with this kind of structure is designated as $\mathbb{Q}$, whose elements are called quaternions. And the mapping $(q,p)\rightarrow qp$ is called the product of q to the right by p (or the product of p to the left by q). The components $\boldsymbol{q}$ and q_0 of a quaternion are considered as its imaginary part $\mathrm{Im}(q)$ and its real part $\mathrm{Re}(q)$ respectively. Unlike complex numbers (in which the imaginary part of $z=x+\boldsymbol{i}y$ is the real number y), the imaginary part of q is effectively a vector in $\mathbb{R}^3$.

The quaternion $q=(-\boldsymbol{q},q_0)$ is called the conjugate of $q=(\boldsymbol{q},q_0)$. It means that the conjugate of a quaternion q is formed by reversing the sign of its imaginary component. It is denoted as $\tilde{q}$. The mapping $q\rightarrow\tilde{q}$ is an automorphism of the vector space $\mathbb{Q}$, but an antiautomorphism of the algebra structure as $\widetilde{qp}=\tilde{p}\tilde{q}$ for any q and p. In addition, quaternions are measurable in the vector space $\mathbb{Q}$ because

$$q\tilde{q}=\tilde{q}q=(\|\mathrm{Re}(q)\|^2+\|\mathrm{Im}(q)\|^2,0)=\|\mathrm{Re}(q)\|^2+\|\mathrm{Im}(q)\|^2 \tag{5}$$

is a sum of two positive numbers. Therefore, it makes sense to define the norm of a quaternion as the scalar $\|q\|=\sqrt{q\tilde{q}}$. Obviously, $\|q\|=0$ if and only if $q=0$. Furthermore, for any quaternion q and p

$$\|pq\|^2=(pq)(\widetilde{pq})=(pq)(\tilde{q}\tilde{p})=p(q\tilde{q})\tilde{p}=(p\tilde{p})\|q\|^2=\|p\|^2\|q\|^2 \tag{6}$$

The formula $q\tilde{q}=\tilde{q}q=\|q\|^2$ has an important consequence. It exhibits explicitly the inverse of a nonzero quaternion, namely $q^{-1}=q/\|q\|^2$, $q\neq0$. The means $\mathbb{Q}$ is a division ring or skew field, an algebraic structure with all the properties of a field except commutativity. Of a quaternion q such that $\|q\|=1$, it is said that it is a unit quaternion. Although the set $\mathbb{Q}$ itself is not a group because the quaternion 0 has no multiplicative inverse, the set $\mathbb{E}=\{q\in\mathbb{Q}\mid \|q\|=1\}$ of unit quaternions is a group. It is not empty: $1\in\mathbb{E}$. If $q\in\mathbb{E}$, so does $\tilde{q}$. In fact, $\tilde{q}$ is just the inverse of q. If p and $q\in\mathbb{Q}$, so does their product pq. It should be noted that the group $\mathbb{E}$ is not commutative.

The aim of this subsection is to show that $\mathbb{E}$ is intimately related to the group $\boldsymbol{SO}(3)$ of rotations in $\mathbb{R}^3$. In the following content, we will donate the unit quaternion as $\varepsilon=(\boldsymbol{\varepsilon},\varepsilon_0)=(\varepsilon_1\ \varepsilon_2\ \varepsilon_3\ \varepsilon_0)$.

To represent a rotation with the unit quaternion $\boldsymbol{\varepsilon}$, a theorem is introduced here. Let $\mathrm{Re}(\varepsilon)=\cos(\omega/2)$ and $\mathrm{Im}(\varepsilon)=\boldsymbol{n}\sin(\omega/2)$, where $\boldsymbol{n}$ is a unit vector. For any $\boldsymbol{x}\in\mathbb{R}^3$, the product $\varepsilon\boldsymbol{x}\tilde{\varepsilon}$ belongs to $\mathbb{R}^3$. Moreover, the application $\boldsymbol{x}\rightarrow\varepsilon\boldsymbol{x}\tilde{\varepsilon}:\mathbb{R}^3\rightarrow\mathbb{R}^3$ is identical to the rotation $\boldsymbol{R}(\omega,\boldsymbol{n})$. The theorem can be proved by calculating that

$$\begin{cases}\varepsilon\boldsymbol{x}=\left(\boldsymbol{x}\cos\dfrac{\omega}{2}+(\boldsymbol{n}\times\boldsymbol{x})\sin\dfrac{\omega}{2},(-\boldsymbol{n}\cdot\boldsymbol{x})\sin\dfrac{\omega}{2}\right)\\ \varepsilon\boldsymbol{x}\tilde{\varepsilon}=\Big\{(\boldsymbol{x}\cdot\boldsymbol{n})\boldsymbol{n}+(\boldsymbol{n}\times\boldsymbol{x})\sin\omega+[(\boldsymbol{n}\times\boldsymbol{x})\times\\ \qquad\boldsymbol{n}]\cos\omega,0\Big\}\end{cases}\tag{7}$$

Eq. (7) is consistent with Eq. (2). Note that both ε and $-\varepsilon$ determine the same transformation, which is related to the appearance of the half angle $\omega/2$. Computationally, it is now an easy matter to use unit quaternions to model rotations. More exciting discussion about quaternions could be found in Refs. [23-24].

2.2 Forward kinematics equations

For a general Stewart mechanism we assume two reference frames separately fixed to the mobile platform and the base. The frame fixed to the mobile platform is called the moving coordinate frame and denoted as $O'\text{-}x'y'z'$. The remaining one is called the static coordinate frame and denoted as $O\text{-}xyz$. The position vectors (also known as location vectors or radius vectors), which determine the position of every spherical joint relative to $O'\text{-}x'y'z'$, are denoted as $\boldsymbol{a}_1,\cdots,\boldsymbol{a}_6$. And the position vectors, which determine the position of every universal joint with respect to $O\text{-}xyz$, are denoted as $\boldsymbol{b}_1,\cdots,\boldsymbol{b}_6$. The position and orientation of $O'\text{-}x'y'z'$ relative to $O\text{-}xyz$ are denoted as $\boldsymbol{P}$ and $\boldsymbol{R}$ respectively. Now we get six equations as follows

$$L_i\boldsymbol{e}_i=\boldsymbol{P}+\boldsymbol{R}\boldsymbol{a}_i-\boldsymbol{b}_i\quad i=1,\cdots,6\tag{8}$$

where the length of the rod i is denoted as L_i and $\boldsymbol{e}_i$, the unit vector in $\mathbb{R}^3$, represents the orientation of the rod i with respect to $O\text{-}xyz$. Considering Eq. (7), Eq. (8) can be extended and expressed in $\mathbb{E}$

$$L_i(\boldsymbol{e}_i,0)=(\boldsymbol{P},0)+\varepsilon(\boldsymbol{a}_i,0)\tilde{\varepsilon}-(\boldsymbol{b}_i,0)\qquad i=1,\cdots,6\tag{9}$$

By making the product to the right with ε for Eq. (9), we get

$$L_i(\boldsymbol{e}_i,0)\varepsilon=(\boldsymbol{P},0)\varepsilon+\varepsilon(\boldsymbol{a}_i,0)-(\boldsymbol{b}_i,0)\varepsilon\qquad i=1,\cdots,6\tag{10}$$

which can be written in the simplified form as

$$L_if_i=h+\varepsilon(\boldsymbol{a}_i,0)-(\boldsymbol{b}_i,0)\varepsilon\quad i=1,\cdots,6\tag{11}$$

when we assume $f_i=(\boldsymbol{e}_i,0)\varepsilon\in\mathbb{E}$ and $h=(\boldsymbol{P},0)\varepsilon=(\varepsilon_0\boldsymbol{P}-\boldsymbol{\varepsilon}\times\boldsymbol{P},-\boldsymbol{\varepsilon}\cdot\boldsymbol{P})\in\mathbb{Q}$.

If we calculate a product to the right by its conjugate quaternion $\widetilde{L_if}$, then we will obtain that

$$L_i^2=\|h\|^2+2\mathrm{Re}\{\tilde{h}[\varepsilon(\boldsymbol{a}_i,0)-(\boldsymbol{b}_i,0)\varepsilon]+(\boldsymbol{a}_i,0)\tilde{\varepsilon}\cdot(\boldsymbol{b}_i,0)\varepsilon\}+\boldsymbol{a}_i\cdot\boldsymbol{a}_i+\boldsymbol{b}_i\cdot\boldsymbol{b}_i\qquad i=1,\cdots,6\tag{12}$$

To achieve a further simplified form we artificially define that $\boldsymbol{A}_i=\boldsymbol{a}_i-\boldsymbol{b}_i$ and $\boldsymbol{B}_i=\boldsymbol{a}_i+\boldsymbol{b}_i$, whose coordinate representations are $(A_{ix}\ A_{iy}\ A_{iz})^{\mathrm{T}}$ and $(B_{ix}\ B_{iy}\ B_{iz})^{\mathrm{T}}$. We define $\boldsymbol{L}=(L_1\ L_2\ L_3\ L_4\ L_5\ L_6)^{\mathrm{T}}$ and $h=(\boldsymbol{h}^{\mathrm{T}},h_0)=(h_1\ h_2\ h_3\ h_0)$, $\boldsymbol{h}\in\mathbb{R}^3$. Eq. (12) can be transformed into the following equations

$$\|h\|^2+2(\boldsymbol{h}\times\boldsymbol{\varepsilon})\cdot\boldsymbol{B}_i+2\boldsymbol{A}_i\cdot(\varepsilon_0\boldsymbol{h}-h_0\boldsymbol{\varepsilon})+\frac{1}{2}(A_{xi}^2-A_{yi}^2-B_{xi}^2+B_{yi}^2)(\varepsilon_1^2-\varepsilon_2^2)+\frac{1}{2}(A_{zi}^2-B_{zi}^2)(2\varepsilon_3^2-\varepsilon_1^2-\varepsilon_2^2)+2\varepsilon_0\boldsymbol{\varepsilon}\cdot(\boldsymbol{B}_i\times\boldsymbol{A}_i)+2\varepsilon_1\varepsilon_2(A_{xi}A_{yi}-B_{xi}B_{yi})+2\varepsilon_2\varepsilon_3(A_{yi}A_{zi}-B_{yi}B_{zi})+2\varepsilon_1\varepsilon_3(A_{xi}A_{zi}-B_{xi}B_{zi})+\frac{1}{2}(\boldsymbol{B}_i^2-\boldsymbol{A}_i^2)(\varepsilon_3^2-\varepsilon_0^2)+\frac{1}{2}(\boldsymbol{A}_i^2+\boldsymbol{B}_i^2)-L_i^2=0\quad i=1,\cdots,6\tag{13}$$

The left side of Eq. (13) can be succinctly written as

$$f_i(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_i\boldsymbol{x}-C_i \quad i=1,\cdots,6 \tag{14}$$

where $\boldsymbol{x}=(\varepsilon_1\ \varepsilon_2\ \varepsilon_3\ \varepsilon_0\ h_1\ h_2\ h_3\ h_0)^{\mathrm{T}}$, $C_i=L_i^2-\frac{1}{2}(\boldsymbol{A}_i^2+\boldsymbol{B}_i^2)$, $i=1,\cdots,6$, and

$\boldsymbol{Q}_i=$

$$\begin{pmatrix}
\boldsymbol{A}_i^2-2A_{ix}^2-\boldsymbol{B}_i^2+2B_{iy}^2 & -2A_{ix}A_{iy}+2B_{ix}B_{iy} & -2A_{ix}A_{iz}+2B_{ix}B_{iz} & 2A_{iz}B_{iy}-2A_{iy}B_{iz} & 0 & 2B_{iz} & -2B_{iy} & 2A_{ix} \\
-2A_{ix}A_{iy}+2B_{ix}B_{iy} & \boldsymbol{A}_i^2-2A_{iy}^2-\boldsymbol{B}_i^2+2B_{iz}^2 & -2A_{iy}A_{iz}+2B_{iy}B_{iz} & -2A_{iz}B_{ix}+2A_{ix}B_{iz} & -2B_{iz} & 0 & 2B_{ix} & 2A_{iy} \\
-2A_{ix}A_{iz}+2B_{ix}B_{iz} & -2A_{iy}A_{iz}+2B_{iy}B_{iz} & \boldsymbol{A}_i^2-2A_{iz}^2-\boldsymbol{B}_i^2+2B_{iz}^2 & 2A_{iy}B_{ix}-2A_{ix}B_{iy} & 2B_{iy} & -2B_{ix} & 0 & 2A_{iz} \\
2A_{iz}B_{iy}-2A_{iy}B_{iz} & -2A_{iz}B_{ix}+2A_{ix}B_{iz} & 2A_{iy}B_{ix}-2A_{ix}B_{iy} & -\boldsymbol{A}_i^2+\boldsymbol{B}_i^2 & -2A_{ix} & -2A_{iy} & -2A_{iz} & 0 \\
0 & -2B_{iz} & 2B_{iy} & -2A_{ix} & -2 & 0 & 0 & 0 \\
2B_{iz} & 0 & -2B_{ix} & -2A_{iy} & 0 & -2 & 0 & 0 \\
-2B_{iy} & 2B_{ix} & 0 & -2A_{iz} & 0 & 0 & -2 & 0 \\
2A_{ix} & 2A_{iy} & 2A_{iz} & 0 & 0 & 0 & 0 & -2
\end{pmatrix}$$

$\boldsymbol{Q}_i$ is a constant symmetric matrix only determined by the structural parameters of the Stewart mechanism.

Additionally, there exit two more equations according to the properties of quaternions

$$\begin{cases} f_7(\boldsymbol{x})=\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2+\varepsilon_0^2-1=0 \\ f_8(\boldsymbol{x})=\varepsilon_1h_1+\varepsilon_2h_2+\varepsilon_3h_3+\varepsilon_0h_0=0 \end{cases} \tag{15}$$

$f_7(\boldsymbol{x})$ and $f_8(\boldsymbol{x})$ can be expressed in the same form as Eq. (14)

$$\begin{cases} f_7(\boldsymbol{x})=\dfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}\begin{pmatrix} 2\boldsymbol{I}_{4\times4} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}_{4\times4} \end{pmatrix}\boldsymbol{x}-1 \\ f_8(\boldsymbol{x})=\dfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}\begin{pmatrix} \boldsymbol{0}_{4\times4} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0}_{4\times4} \end{pmatrix}\boldsymbol{x} \end{cases} \tag{16}$$

The unknown $\boldsymbol{x}$ can be changed into study coordinates, a dual quaternion or a biquaternion by a linear transformation. In this way Eqs. (14, 16) will have a little change but the quadratic form is unchanged.

Eqs. (13,15) constitute eight quadratic polynomial equations, by dealing with which we will obtain the solution to the forward kinematics problem.

3 Fast Forward Kinematics Method

3.1 Construction of iterative sequence

For any of Eqs. (13, 15), if let $\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^8$, then

$$f_i(\boldsymbol{a})-f_i(\boldsymbol{b})=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{Q}_i(\boldsymbol{a}-\boldsymbol{b})+\frac{1}{2}(\boldsymbol{a}-\boldsymbol{b})^{\mathrm{T}}\boldsymbol{Q}_i(\boldsymbol{a}-\boldsymbol{b}) \quad i=1,\cdots,8 \tag{17}$$

We assume $\boldsymbol{x}^*\in\mathbb{R}^8$ is a real solution to those quadratic equations and that $\boldsymbol{x}_k\in\mathbb{R}^8$ is an approximation of $\boldsymbol{x}^*$. In Eq. (17) we plug in $\boldsymbol{a}=\boldsymbol{x}^*$, $\boldsymbol{b}=\boldsymbol{x}_k$, $\Delta\boldsymbol{x}=\boldsymbol{x}^*-\boldsymbol{x}_k$ and omit $(\Delta\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_i\Delta\boldsymbol{x})/2$ which is the second-order trace of $\Delta\boldsymbol{x}$. In the geometric sense it means replacing the quadric surface with the tangent plane at $\boldsymbol{x}_i$. Since $f_i(\boldsymbol{x}^*)=0$, we will get

$$-f_i(\boldsymbol{x}_k)\approx\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{Q}_i(\boldsymbol{x}^*-\boldsymbol{x}_k) \quad i=1,\cdots,8 \tag{18}$$

Therefore, we get the iterative sequence written as

$$\boldsymbol{x}_{k+1}=\Phi(\boldsymbol{x}_k)=\boldsymbol{x}_k-\boldsymbol{J}_k^{-1}\boldsymbol{F}(\boldsymbol{x}_k) \quad k=0,1,2,\cdots \tag{19}$$

where $\boldsymbol{J}_k=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_i\ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_2\ \cdots\ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_8)^{\mathrm{T}}$ and $\boldsymbol{F}(\boldsymbol{x}_k)=(f_1(\boldsymbol{x}_k)\ f_2(\boldsymbol{x}_k)\cdots f_8(\boldsymbol{x}_8))^{\mathrm{T}}$.

Eq. (19) is the general form of Newton-Raphson method. Furthermore, the method can be simplified nicely when applied to a system of quadratic equations.

Considering the following relationship between $\boldsymbol{F}(\boldsymbol{x}_k)$ and $\boldsymbol{J}_k$

$$\boldsymbol{F}(\boldsymbol{x}_k)=\frac{1}{2}\boldsymbol{J}_k\boldsymbol{x}_k-\boldsymbol{C}$$

$$\boldsymbol{C}=(C_1\quad C_2\quad\cdots\quad C_6\quad 1\quad 0)^{\mathrm{T}} \tag{20}$$

We eliminate $\boldsymbol{J}_k^{-1}\boldsymbol{F}(\boldsymbol{x}_k)$ in Eq. (19) and the iterative function can be expressed in such a form as

$$\Phi(\boldsymbol{x}_k)=\frac{1}{2}\boldsymbol{x}_k+\boldsymbol{J}_k^{-1}\boldsymbol{C} \tag{21}$$

Now it need not to calculate $\boldsymbol{F}(\boldsymbol{x}_k)$ in each step of iteration, which saves some computation and is essential for real-time applications. Since calculating the inverse of $\boldsymbol{J}_k$ is time-consuming, we replace it with solving a system of linear equations with a numerical method in the actual calculation and adopt the following iterative sequence

$$\begin{cases} \boldsymbol{x}_{k+1}=\dfrac{1}{2}\boldsymbol{x}_k+\Delta\boldsymbol{x}_k \\ \boldsymbol{J}_k\Delta\boldsymbol{x}_k=\boldsymbol{C} \end{cases} \quad k=0,1,2,\cdots \tag{22}$$

3.2 Convergence: singularity and initial guess

It is well known that Newton-Raphson method converges quadratically as long as the Jacobian matrix of the system is nonsingular and the initial guess is close enough to a solution.

When the Jacobian matrix is singular or near singularities, the iterative solution can jump far from the local solution and fail to converge or converge to a different solution. To avoid the situation, we make a bit change in Eq. (22). We have known that if $\boldsymbol{J}_k$ is replaced by a constant matrix $\boldsymbol{J}_0$ in Eq. (22), the resulting iterative sequence converges linearly and the Jacobian matrix will not become singular during iterations. This kind of iterative method is named as simplified Newton-Raphson method.

Now we combine the general Newton-Raphson method with the simplified Newton-Raphson method. If we predict the Jacobian matrix $\boldsymbol{J}_k$ is very close to singularity during the iteration of step k, the simplified Newton-Raphson method is used in this step

$$\begin{cases}\boldsymbol{x}_{k+1}=\dfrac{1}{2}\boldsymbol{x}_k+\Delta\boldsymbol{x}_k\\ \boldsymbol{J}_{k-1}\Delta\boldsymbol{x}_k=\boldsymbol{C}\end{cases}\tag{23}$$

Many characteristics can help to judge the singularity of the Jacobian matrix, such as the large condition number, the small determinant, and the large change of $\Delta\boldsymbol{x}_k$. Obviously, we calculate the change of $\Delta\boldsymbol{x}_k$ since it costs less time than the other two ways. In the next iteration we return back to Eq. (22). In this way we avoid the singularity at the expense of a little computing time.

How to choose the initial guess is also very important. According to the local convergence theorem[25], there exists a neighborhood of $\boldsymbol{x}^*$ denoted as $S_\delta=\{\boldsymbol{x}\in\mathbb{R}^8 \mid \|\boldsymbol{x}^*-\boldsymbol{x}\|<\delta\}$, where Eq. (22) will converge to $\boldsymbol{x}^*$ with at least square convergence for any initial guess $\boldsymbol{x}_0\in S_\delta$.

On the other hand, the rod length $\boldsymbol{L}$ is a continuous function of $\boldsymbol{x}$, so that there exists a neighborhood of the actual $\boldsymbol{L}^*$ denoted as $T_\tau=\{\boldsymbol{L}\in\mathbb{R}^6 \mid \|\boldsymbol{L}^*-\boldsymbol{L}\|<\tau\}$ and if $\boldsymbol{L}\in T_\tau$, $\boldsymbol{x}\in S_\delta$. Thus, the change of the pose of the mobile platform should stay in the permitted range by controlling the change of the rod lengths. In general applications when the mobile platform moves continuously at a given requirement, the rod lengths change over time continuously. $\boldsymbol{L}$ is a time-dependent function denoted as $\boldsymbol{L}(t)$. With time ranging from t_0 to t, we divide the time segment $\Delta t=t-t_0$ into several pieces called the control cycles. During each cycle the pose of the mobile platform is figured out by Eq. (22) where the initial value of Eq. (22) is the result calculated in the previous cycle. All of these processes help to ensure the convergence of the iterative sequence.

4 Numerical Simulation

The algorithm derived in Section 3 is applicable to a general Stewart mechanism with any configuration. Without loss of generality, a kind of Stewart mechanism with the common configuration, where the universal joints and spherical joints are distributed symmetrically in two circles as shown in Fig. 2, is used as an example to verify the effectiveness of the algorithm.

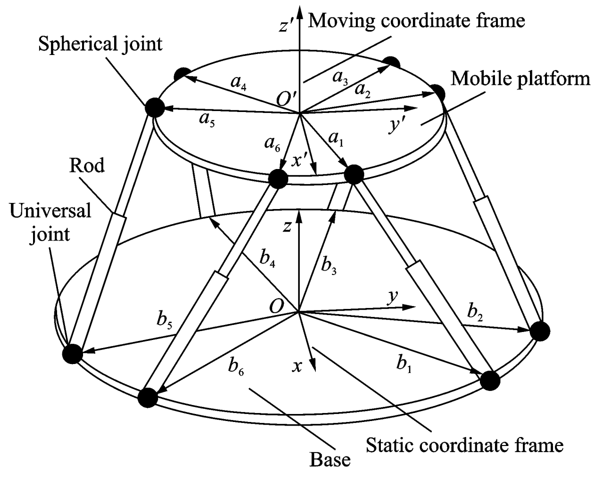


Fig. 2 Commonly used Stewart mechanism

The coordinate presentations of the vectors that describe the connecting positions of two platforms with six rods are expressed as follows

$$\boldsymbol{a}_{2i-1}=\begin{pmatrix}\cos\left(\dfrac{2\pi}{3}(i-1)+\dfrac{\pi}{12}\right)\\ \sin\left(\dfrac{2\pi}{3}(i-1)+\dfrac{\pi}{12}\right)\\ 0\end{pmatrix}\quad i=1,2,3$$

$$\boldsymbol{a}_{2i}=\begin{pmatrix}\cos\left(\frac{2\pi}{3}i-\frac{\pi}{12}\right)\\ \sin\left(\frac{2\pi}{3}i-\frac{\pi}{12}\right)\\ 0\end{pmatrix}\quad i=1,2,3$$

$$\boldsymbol{b}_{2i-1}=2\begin{pmatrix}\cos\left(\frac{2\pi}{3}(i-1)+\frac{\pi}{6}\right)\\ \sin\left(\frac{2\pi}{3}(i-1)+\frac{\pi}{6}\right)\\ 0\end{pmatrix}\quad i=1,2,3$$

$$\boldsymbol{b}_{2i}=2\begin{pmatrix}\cos\left(\frac{2\pi}{3}i-\frac{\pi}{6}\right)\\ \sin\left(\frac{2\pi}{3}i-\frac{\pi}{6}\right)\\ 0\end{pmatrix}\quad i=1,2,3$$

In dynamic simulation, the starting pose of Stewart mechanism is known and serves as the initial value of the algorithm. During each sampling period T the algorithm searches for the new solution, which becomes the initial guess in the next cycle. The movement is defined as a time-dependent function of the pose of the mobile platform. The position is denoted by the vector $\boldsymbol{P}$ and the orientation is denoted by the unit quaternion ε, therefore, the movement can be define with

$$\begin{cases}\boldsymbol{P}(t)=\left(0.1\sin\frac{\pi t}{2}0.12\sin\frac{\pi t}{2}1+0.15\sin t\right)^{\mathrm{T}}\\ \varepsilon(t)=\boldsymbol{n}\sin\frac{\omega}{2}\cos\frac{\omega}{2}\end{cases}$$
$$0\leqslant t\leqslant 2 \tag{24}$$

where $\omega=\frac{\pi}{4}\sin(2\pi t)$, $\boldsymbol{n}=(\sin\gamma\cdot\cos\alpha\sin\gamma\cdot\sin\alpha\cos\gamma)$, $\gamma=\frac{\pi}{12}\sin\left(2\pi t+\frac{\pi}{2}\right)+\frac{5\pi}{12}$, $\alpha=2\pi\sin(2\pi t)$. The starting pose of the mobile platform is $\boldsymbol{P}(t=0)=(0\ 0\ 1)^{\mathrm{T}}$ and $\varepsilon(t=0)=(0\ 0\ 0\ 1)^{\mathrm{T}}$.

The sampling period T is set as 1 ms, which equals a 1 000 Hz sampling frequency. We make a dynamic simulation of 1 000 ms. The machine precision of the computer is set as 16. Obviously, we will get different calculation accuracies with different times of iterations.

The curves shown in Figs. 3,4 represent the absolute difference between the calculated and the actual poses with 10 times of iterations. Due to the large number of cycles, the calculation error is defined as the largest absolute difference in the last 100 ms, therefore, the graphs in each point show the worst case in the last 100 ms of the simulation. The errors are presented separately for positions and orientations of the mobile platform. Fig. 5 shows the influence of the times of iterations on calculation accuracy. If we take 10 times of iterations, we will get a calculation error of 3.11×10^{-15}. And we get an error of 1.46×10^{-7} for 2 iterations. With the number of iterations greater than 4, the error magnitude downs to 10^{-15}. It cannot go smaller with more iterations because it is restricted to the machine precision which is set as 16. In real-time applications, we can choose the times of iterations based on the re-

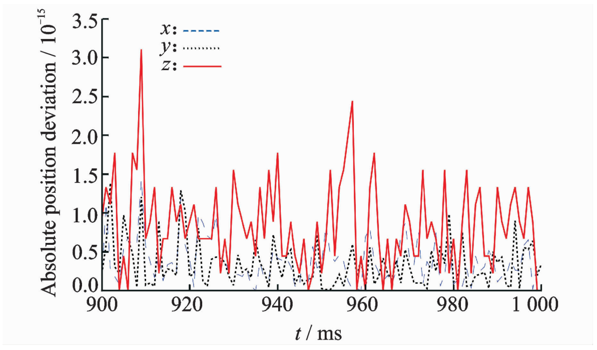


Fig. 3 Absolute position deviation

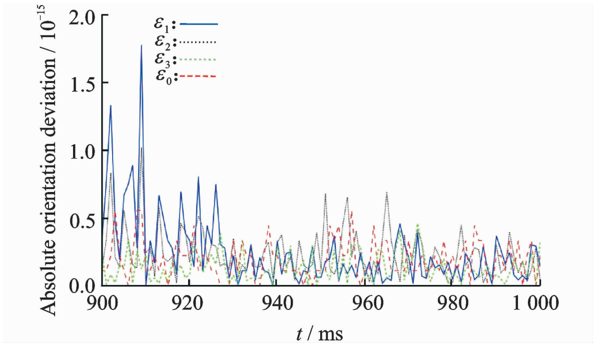


Fig. 4 Absolute orientation deviation

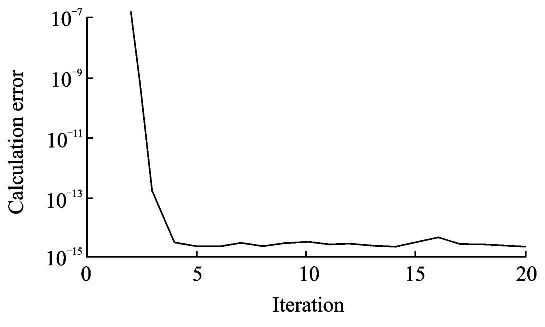


Fig. 5 Influence of times of iterations on calculation accuracy

quired precision.

We have attempted a kind of widely used algorithm "FindRoot" provided by a software-Wolfram Mathematica 9.0 to solve the forward kinematics equations with a classic form, where the Euler angles are used to describe the rotation, and find that our new algorithm saves 43.92% computing time with 5 times of iteration as showed in Tables 1, 2. The initial guess in Table 2 is just the result in Table 1. Our new algorithm costs 0.352 562 ms while the Mathematica′s method costs 0.628 684 ms.

Table 1 One process of our new algorithm

Time of iteration	New algorithm							
	ε_1	ε_2	ε_3	ε_0	h_1	h_2	h_3	h_0
0	0	0	0	1	0	0	1	0
1	0.041 007 2	0.014 534 8	0.014 534 8	1	0.084 32	0.011 93	1.632 8	0.017 208 3
2	0.027 148 3	0.009 628 32	−0.017 216 1	0.999 545	0.086 462 4	0.011 071 2	1.506 41	0.023 464
3	0.026 024 2	0.009 181 15	−0.017 217 4	0.999 472	0.086 692 6	0.010 769 6	1.500 92	0.023 499 1
4	0.026 019 7	0.009 179 05	−0.017 217 4	0.999 471	0.086 695	0.010 767 2	1.500 9	0.023 499 4
5	0.026 019 7	0.009 179 05	−0.017 217 4	0.999 471	0.086 695	0.010 767 2	1.500 9	0.023 499 4

Table 2 Another process of our new algorithm

Time of iteration	New algorithm							
	ε_1	ε_2	ε_3	ε_0	h_1	h_2	h_3	h_0
0	0.026 019 7	0.009 179 05	−0.017 217 4	0.999 471	0.086 695	0.010 767 2	1.500 9	0.023 499 4
1	0.018 602 5	0.014 436 3	−0.017 141 4	0.999 617	0.002 625 63	−0.002 192 54	1.099 75	0.019 365 8
2	0.017 197 8	0.017 456 7	−0.017 142 4	0.999 558	0.002 282 73	−0.002 162 39	1.023 14	0.017 546 1
3	0.017 142 6	0.017 750 6	−0.017 142 5	0.999 549	0.002 227 51	−0.002 162 72	1.020 24	0.017 497 7
4	0.017 142 6	0.017 751 6	−0.017 142 6	0.999 548	0.002 227 16	−0.002 162 72	1.020 24	0.017 497 6
5	0.017 142 6	0.017 751 6	−0.017 142 6	0.999 548	0.002 227 16	−0.002 162 72	1.020 24	0.017 497 6

5 Conclusions

The property of quaternion is studied to describe the three-dimensional rotation and model the forward kinematics of the general Stewart mechanism. The forward kinematics equations are expressed as a system of quadratic equations with a simple and symmetric form. Newton-Raphson method simplifies nicely when applies to this system, which is good for the real-time control.

In addition, the singularity of the Jacobian matrix is discussed and how to determine the initial guess of the algorithm. Some techniques are proposed and integrated to ensure the convergence and stability. In order to verify the efficiency of the algorithm, we carry out a dynamic simulation of 1 kHz sampling frequency and the result shows that with 4 times of iterations the calculation error magnitude downs to 10^{-15} which reaches the machine precision of the computer.

The numerical algorithm has the advantage of high accuracy and fast convergence speed that are sorely required for real-time applications. It can be applied in the feedback motion control of the Stewart mechanism to achieve the multiple input multiple output control based on task space, which will improve the control precision. Such an approach and the algorithm may be extended to other kinds of mechanisms.

References:

[1] Merlet J P. Parallel robots[M]. 2nd Edition. Dordrecht, The Netherlands: Springer, 2006.

[2] Huang X, Liao Q, Wei S. Closed-form forward kinematics for a symmetrical 6-6 Stewart platform using algebraic elimination[J]. Mechanism and Machine Theory, 2010,45(2):327-334.

[3] Gan D, Liao Q, Dai J S, et al. Forward displacement analysis of the general 6-6 Stewart mechanism using Gröbner bases[J]. Mechanism and Machine Theory,

2009,44(9):1640-1647.

[4] Innocenti C. Forward kinematics in polynomial form of the general Stewart platform[J]. Journal of Mechanical Design, 2001,123(2):254-260.

[5] Lee T Y, Shim J K. Forward kinematics of the general 6-6 Stewart platform using algebraic elimination [J]. Mechanism and Machine Theory, 2001,36(9): 1073-1085.

[6] Huang X G. An approach for direct kinematics of a parallel manipulator robot[M]//Information and Automation. Berlin Heidelberg: Springer, 2011: 524-528.

[7] Akcali I D, Mutlu H. A novel approach in the direct kinematics of Stewart platform mechanisms with planar platforms [J]. Journal of Mechanical Design, 2006,128(1):252-263.

[8] Cheng Shili, Wu Hongtao, Wang Chaoqun, et al. Forward kinematics analysis of 6-3 Stewart parallel mechanisms based on orthogonal complement method [J]. China Mechanical Engineering, 2011, 22(5): 505-509. (in Chinese)

[9] Cheng Shili, Wu Hongtao, Yao Yu, et al. An analytical method for the forward kinematics analysis of 6-SPS parallel mechanisms[J]. Chinese Journal of Mechanical Engineering, 2010, 46(9): 26-31. (in Chinese)

[10] Chiu Y J, Perng M H. Forward kinematics of a general fully parallel manipulator with auxiliary sensors [J]. The International Journal of Robotics Research, 2001,20(5):401-414.

[11] Parenti-Castelli V, Di Gregorio R. A new algorithm based on two extra-sensors for real-time computation of the actual configuration of the generalized Stewart-Gough manipulator[J]. Journal of Mechanical Design, 2000,122(3):294-298.

[12] Yang C, Zheng S, Jin J, et al. Forward kinematics analysis of parallel manipulator using modified global Newton-Raphson method [J]. Journal of Central South University of Technology, 2010,17:1264-1270.

[13] Liu K, Lewis F L, Fitzgerald M. Solution of nonlinear kinematics of a parallel-link constrained Stewart platform manipulator[J]. Circuits, Systems and Signal Processing, 1994,13(2/3):167-183.

[14] Merlet J P. Direct kinematics of parallel manipulators [J]. IEEE Transactions on Robotics and Automation, 1993,9(6):842-846.

[15] Parikh P J, Lam S S Y. A hybrid strategy to solve the forward kinematics problem in parallel manipulators[J]. IEEE Transactions on Robotics, 2005,21 (1):18-25.

[16] He J, Gu H, Wang Z. Solving the forward kinematics problem of six-DOF Stewart platform using multitask Gaussian process[J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2013,227(1):161-169.

[17] Rolland L, Chandra R. On solving the forward kinematics of the 6-6 general parallel manipulator with an efficient evolutionary algorithm[M]//ROMANSY 18 Robot Design, Dynamics and Control. Vienna: Springer, 2010:117-124.

[18] Ghasemi A, Eghtesad M, Farid M. Neural network solution for forward kinematics problem of cable robots[J]. Journal of Intelligent & Robotic Systems, 2010,60(2):201-215.

[19] Omran A, Bayoumi M, Kassem A, et al. Optimal forward kinematics modeling of Stewart manipulator using genetic algorithms[J]. Jordan Journal of Mechanical & Industrial Engineering, 2009,3(4):280-293.

[20] Parikh P J, Lam S S. Solving the forward kinematics problem in parallel manipulators using an iterative artificial neural network strategy[J]. The International Journal of Advanced Manufacturing Technology, 2009,40(5/6):595-606.

[21] Wang Z, He J, Shang H, et al. Forward kinematics analysis of a six-DOF Stewart platform using PCA and NM algorithm[J]. Industrial Robot: An International Journal, 2009,36(5):448-460.

[22] Selig J M. Geometric fundamentals of robotics[M]. 2nd Edition. NY, USA: Springer, 2005:16-18.

[23] Arribas M, Elipe A, Palacios M. Quaternions and the rotation of a rigid body[J]. Celestial Mechanics and Dynamical Astronomy, 2006,96(3/4):239-251.

[24] Hanson A J. Visualizing quaternions [M]. NY, USA: Morgan Kaufmann, 2005:46-56.

[25] Burden R L, Douglas Faires J. Numerical analysis [M]. 9th Edition. Boston, USA: PWS Publishing Company, 2011:31-36.

(Executive editor: Xu Chengting)