ORIGINAL ARTICLE

Open Access

CrossMark

Design and Analysis for a Three-Rotational-DOF Flight Simulator of Fighter-Aircraft

Chang-Chun Zhou and Yue-Fa Fang[*]

Abstract

Most of researchers focused on traditional six degrees of freedom (DOF) Stewart flight simulator, which can not be adaptive in fighter-aircraft flight simulator. A three rotational DOF flight simulator of fighter-aircraft based on double parallel manipulator and hybrid structure is presented. The flight simulator is composed of two identical 3-RRS (revolute-revolute-spherical) spherical parallel manipulators and one cabin, called Twins. The cabin has an additional independent DOF for 360° continuous rotation, so it can be applied as a flight simulator for a fighter-aircraft to achieve spin maneuvering. Because of the introduction of the hybrid structure and double parallel manipulator of the mechanism, the redundancy exists with respect to both kinematics and actuation. Kinematics is carried out and Jacobian matrix is established by means of screw theory. The inverse kinematics is given out by the analytical method. 64 groups inverse solutions are showed in a table by permutation. Forward kinematics is solved by an effectively numerical method. The forward numerical method is realized based on the analytically inverse kinematics and Jacobian matrix. The numerical examples show that the forward numerical method can be used in real-time control. The rolling motion is considered in forward kinematics and a numerical example is given out. The proposed flight simulator can spin and there are three rotational DOF with a hybrid structure so that the novel flight simulator can be used in the field of the fighter-aircraft for pilots to train.

Keywords: Parallel manipulator, Hybrid mechanisms, Actuation redundancy, Flight simulator, Kinematics, Screw theory, Jacobian matrix

1 Introduction

Flight simulators are devices in which air crews and pilots can train without the use of an actual aircraft [1]. In flight training, they are used mainly to reduce costs and increase safety. In their most sophisticated form, they simulate an aircraft's vehicular motion, instrumentation and sounds, gravitational forces, radar and electro-optical sensor displays, and out-the-window views. According to the USA's Federal Aviation Administration (FAA) regulations, any device called a Flight Simulator must have at least one motion platform, otherwise it can only be termed a Flight Training Device [2]. Therefore, the vehicular motion platform in flight simulator is one of the most important parts.

Since Stewart's initial use of a 6-DOF parallel manipulator as a flight simulator in 1965 [3], this approach has become standard. Over the past five decades, the Stewart parallel manipulator has been used to make significant contributions to aeronautical research [4–7]. Even so, there is still the disadvantage of the Stewart-type flight simulator in that its posture rotation range is less than 30°. The limitation of the posture range of a Stewart parallel manipulator hinders its ability to serve as a flight simulator of a fighter-aircraft. The motion of a fighter-aircraft involves continuous 360° rolling frequently. In order to achieve continuous 360° rolling, Kim et al. [8] presented an innovative motion base as a flight simulator, based on a 6-DOF parallel mechanism, called Eclipse-II. The Eclipse-II allows continuous 360° rotation in A, B, and C-axes as well as translational motions in X, Y, and Z-axes. However, the rotations of the Eclipse-II parallel

*Correspondence: yffang@bjtu.edu.cn
Robot Research Center, Beijing Jiaotong University, Beijing 100044, China

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 2 of 12

manipulator are achieved by two circular guide rails, which severely affect the dynamical performance. As a matter of fact, in both the Stewart and Eclipse-II simulators, translational movements are nearly inoperative. Low-cost flight simulator with a reduced-DOF (less than six) platform was proposed by Pouliot and Gosselin [9], which revealed result comparable to those of a 6-DOF Stewart platform. Subsequently, Shui et al. [10] proposed a more advanced and innovative reconfigurable spherical motion generator to enable continuous spherical motion of the flight simulator. The mechanism enables unlimited workspace with respect to 3-DOF spherical motion with rapid, continuous, and precise motion capabilities. However, it is actuated by an electromagnetic motor with a highly complicated configuration that causes the volume of manipulator to be huge. So far, no parallel manipulator can be used as a flight simulator to accomplish a 360° continuous rotation in practice.

Although it is not necessary to be able to continuously rotate in all orientations, a flight simulator of a fighter-aircraft needs to be able to roll in an additional 360° to perform most motions of a fighter-aircraft. Therefore, the purpose of this study is to achieve continuous 360° rotation of a fighter-aircraft flight simulator.

Many researchers have focused on hybrid platforms such as the famous Tricept by Neumann [11] and Trivariant by Huang et al. [12] that provide a good workspace volume. They are typically a combination of a parallel manipulator and a serial manipulator. As a hybrid alternative, a double parallel manipulator was proposed for enlarging the workspace by Lee et al. [13, 14]. Tsai et al. [15] used a double parallel manipulator to enlarge the workspace of the manipulator and analyzed its kinematic properties. Furthermore, many researchers have noted that the redundancy of a double parallel manipulator can improve its abilities and performance, for example, enlarging the volume of the workspace. There are two main types of redundancy for the parallel manipulator: (a) kinematic redundancy and (b) actuation redundancy [16, 17]. Zanganeh and Wang et al. [18–20] studied the kinematic redundancy of a parallel manipulator by analyzing its kinematics, and pointed out that the extra DOF not only allow for execution of the original output task, but also additional tasks such as increasing the workspace. It is notable that the Eclipse-II is a redundant actuation parallel manipulator. Redundant actuation has been proven to be a good method to enhance performance of parallel manipulators. Nokleby et al. [21] proved that redundant actuation can improve force capabilities. Kim et al. [22] investigated the redundantly actuated parallel manipulator and proved that redundant actuation of a parallel manipulator not only can improve force capabilities, but can also enhance the stiffness of a manipulator.

Li et al. [23] derived the conclusion that redundant actuation has little effect on the stiffness when the actuators are on prismatic joints and enhances stiffness value greatly when the actuators are on revolute joints.

Although the novel mechanisms are proposed one by one, they don't adapt for the flight simulator. Regarding existing parallel manipulators, the 3-DOF spherical parallel manipulator is a compact configuration with large rotational posture [24]. Therefore, given the above conclusions, a flight simulator of a fighter-aircraft based on a double and hybrid 3-RRS [25–28] spherical parallel manipulator was chosen for this study.

The remainder of this paper is organized as follows. In Section 2, the structure of the Twins flight simulator is described. In Section 3, the kinematic properties of the Twins are discussed, including the development of the direction-cosine matrixes in Section 3.1 with analytical spherical theory, the analysis of DOF in Section 3.2, and the development of the Jacobian matrix via screw theory in Section 3.3. Inverse kinematics and forward kinematics are conducted in Sections 4 and 5. Numerical examples are provided in Section 6. Lastly, the conclusions are discussed.

2 Architectural Description

As shown in Figure 1, the flight simulator is a hybrid manipulator developed on the basis of two identical 3-RRS spherical parallel mechanisms and one cabin with an independent DOF. Therefore, it has kinematic redundancy and a redundantly actuated manipulator as well. The basic makeup of the whole platform consists of two fixed platforms, one cabin, six RRS legs, two flange-bearing-gear decelerators, and eight motors.
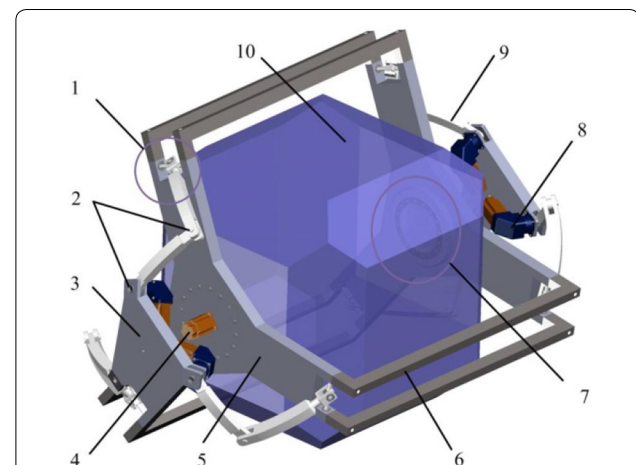

Figure 1 Flight simulator of fighter-aircraft called "Twins". 1. Spherical joint (substituted with one Hook joint and one R joint); 2. R joint; 3. Fixed platform; 4. Motor; 5. Moving platform; 6. Connecting rod; 7. Flange-Bearing-Gear reducer; 8. Gear reducer; 9. Leg; 10. Cabin
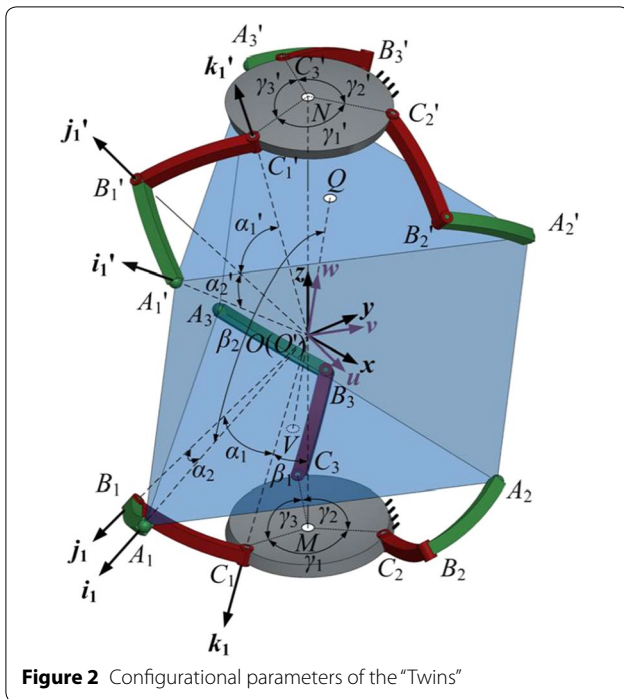
Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 3 of 12

Figure 2 Configurational parameters of the "Twins"

Figure 3 Flange-bearing-gear reducer

As shown in Figure 2, all joints are located on spherical surface and all joint axes intersect at a single point O (center of the sphere). The active joints are C_i and C_i', and the passive joints are B_i, B_i', A_i and A_i', $i=1, 2, 3$. A fixed Cartesian coordinate frame $O-\boldsymbol{xyz}$ and a moving Cartesian coordinate frame $O'-\boldsymbol{uvw}$ are positioned at the center of the sphere. In the fixed coordinate frame, the $\boldsymbol{x}$-axis is normal to plane C_1ON, and extends outwards. The $\boldsymbol{z}$-axis superposes on line MN, and extends upwards. The $\boldsymbol{y}$-axis is defined by the right hand rule. In the moving coordinate frame, the $\boldsymbol{u}$-axis is normal to plane A_1ON, the $\boldsymbol{w}$-axis superposes on line QV (the points Q and V are the geometrical center of triangles $\Delta A_1'A_2'A_3'$ and $\Delta A_1A_2A_3$), and extends outwards. The $\boldsymbol{v}$-axis is defined by right hand rule.

Looking at Figure 2, based on the symmetrical rule [29], $\gamma_1=120°$, $\gamma_2=120°$, $\gamma_3=120°$, $\gamma_1'=120°$, $\gamma_2'=120°$, $\gamma_3'=120°$. α_1 is the angle between the B_i-axis and C_i-axis, and similarly α_1' is the angle between the B_i' axis and C_i' axis. α_2 is the angle between the B_i-axis and A_i-axis, α_2' is the angle between the B_i' axis and A_i' axis. β_1 is the angle between line MN and line OC_i. β_2 is the angle between line QV and OA_i. $0°<\alpha_1=\alpha_1'<90°$, $0°<\alpha_2=\alpha_2'<90°$, $0°<\beta_1<90°$, $0°<\beta_2<90°$.

Looking at Figure 3, to achieve 360° rotation of the cabin, two flange-bearing-gear reducers are used. The outer flange is fixed to the moving platform by bolts,
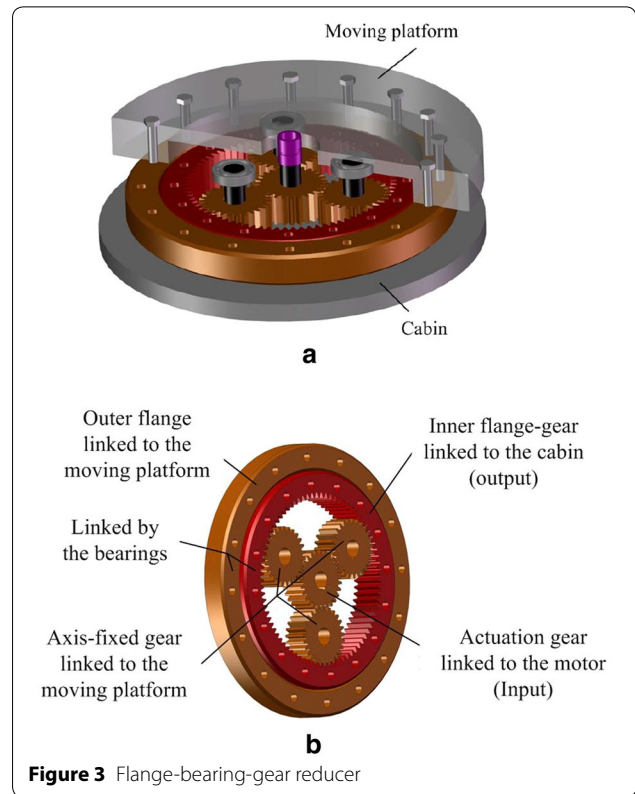
and the inner flange is fixed to the cabin by bolts as well. Together, the outer flange and inner flange serve as a bearing. Three axis-fixed gears are fixed to the moving platform via shafts, and the center gear is linked to a motor.

3 Degrees of Freedom and Jacobian Matrix
3.1 Development of Direction-cosine Matrices
The direction cosines of a vector are the cosines of the angles between the vector and the three coordinate axes. Deriving direction cosines are required in this paper to perform inverse kinematics and obtain the Jacobian matrix. A 3-RRS spherical parallel manipulator and its mirror image are shown in Figure 4.

On the basis of analytical theory of spherical space [30], the direction cosines matrices of B_i, B_i' are built first. The fixed Cartesian coordinate frame of the two 3-RRS spherical parallel manipulators is shown in Figure 4, where the $\boldsymbol{x}$-axis and $\boldsymbol{x}'$-axis are inverses of each other, as are the $\boldsymbol{z}$-axis and $\boldsymbol{z}'$-axis and the $\boldsymbol{y}$-axis and $\boldsymbol{y}'$-axis. The direction cosines matrices of ${}^o\boldsymbol{B}_i$ and ${}^o\boldsymbol{B}_i'$ in the fixed Cartesian coordinate frame O-$\boldsymbol{xyz}$ are as follows:

$$
{}^o\boldsymbol{B}_i = \boldsymbol{R}(\boldsymbol{z}, \phi_i) \begin{pmatrix} \sin\alpha_1 \sin\theta_i \\ -(\sin(\pi-\beta_1)\cos\alpha_1 + \cos(\pi-\beta_1)\sin\alpha_1\cos\theta_i) \\ \cos(\pi-\beta_1)\cos\alpha_1 - \sin(\pi-\beta_1)\sin\alpha_1\cos\theta_i \end{pmatrix}, \tag{1}
$$

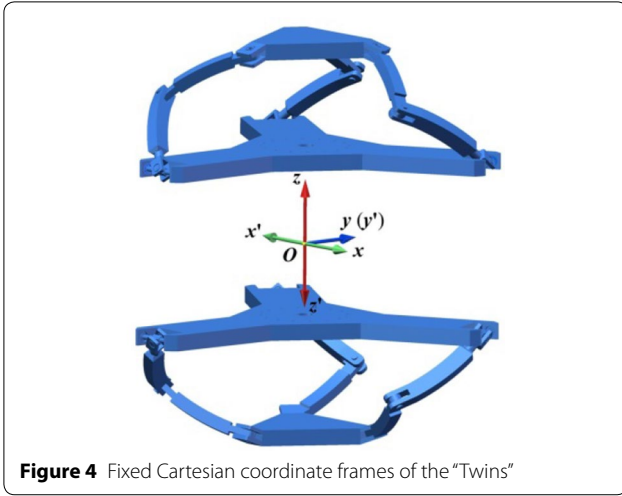Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 4 of 12

Figure 4 Fixed Cartesian coordinate frames of the "Twins"

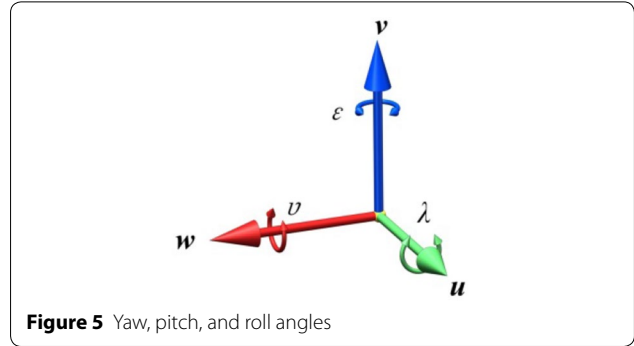**Figure 5** Yaw, pitch, and roll angles

$$^oA_2 = R(z, \varphi_2)^oA_1, \tag{11}$$

$$^oA_3 = R(z, \varphi_2)^oA_3, \tag{12}$$

$$^oB'_i = R(z, \phi_i) \begin{pmatrix} -\sin\alpha'_1\sin\theta'_i \\ -(\sin(\pi-\beta_1)\cos\alpha'_1 + \cos(\pi-\beta_1)\sin\alpha'_1\cos\theta'_1) \\ -\cos(\pi-\beta_1)\cos\alpha'_1 + \sin(\pi-\beta_1)\sin\alpha'_1\cos\theta'_1 \end{pmatrix}, \tag{2}$$

where θ_i and θ'_i are input angles of the C_i-axis and C'_i-axis, and where $\phi_1 = 0°$, $\phi_2 = \gamma_1 = 120°$, and $\phi_3 = \gamma_1 + \gamma_2 = 240°$.

The rotation matrix $R(z, \phi_i)$ with reference to the z-axis is as follows:

$$R(z, \varphi_i) = \begin{pmatrix} \cos\varphi_i & -\sin\varphi_i & 0 \\ \sin\varphi_i & \cos\varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

In the fixed Cartesian coordinate frame, the direction cosines matrices of ^oC_i and $^oC'_i$ can be given directly as follows:

$$^oC_1 = R(z, \varphi_1)(0 \quad -\sin\beta_1 \quad -\cos\beta_1)^T, \tag{4}$$

$$^oC_2 = R(z, \varphi_2)^oC_1, \tag{5}$$

$$^oC_3 = R(z, \varphi_3)^oC_1, \tag{6}$$

$$^oC'_1 = R(z', \varphi_1)(0 \quad -\sin\beta_1 \quad \cos\beta_1)^T, \tag{7}$$

$$^oC'_2 = R(z, \varphi_2)^oC'_1, \tag{8}$$

$$^oC'_3 = R(z, \varphi_3)^oC'_1. \tag{9}$$

In the moving coordinate frame, the direction cosines matrices of A_i and A_i' are expressed as follows:

$$A_1 = R(z, \varphi_1)\begin{pmatrix} 0 & -\sin\beta_2 & \cos\beta_2 \end{pmatrix}^T, \tag{10}$$

$$A'_1 = R(z', \varphi_1)\begin{pmatrix} 0 & -\sin\beta_2 & -\cos\beta_2 \end{pmatrix}^T, \tag{13}$$

$$A'_2 = R(z, \varphi_2)A'_1, \tag{14}$$

$$A'_3 = R(z, \varphi_3)A'_1. \tag{15}$$

In the fixed coordinate frame, ^oA_i and $^oA_i'$ can be depicted as follows:

$$^oA_i = TA_i, i = 1, 2, 3, \tag{16}$$

$$^oA'_i = TA'_i, i = 1, 2, 3, \tag{17}$$

where T is the Euler angle rotation matrix with regard to the moving coordinate frame $O' - uvw$, and is depicted as follows:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\lambda & -\sin\lambda \\ 0 & \sin\lambda & \cos\lambda \end{pmatrix} \begin{pmatrix} \cos\varepsilon & 0 & \sin\varepsilon \\ 0 & 1 & 0 \\ -\sin\varepsilon & 0 & \cos\varepsilon \end{pmatrix} \times \\ \begin{pmatrix} \cos\upsilon & -\sin\upsilon & 0 \\ \sin\upsilon & \cos\upsilon & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{18}$$

where λ, ε, υ are the yaw, pitch, and roll angles (see Figure 5).

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 5 of 12

3.2 Degrees of Freedom Analysis

When the cabin is fixed, the degrees of freedom of the mechanism can be obtained by using the modified formula of Grübler–Kutzbach:

$$M = d(n - g - 1) + \sum_{i=1}^{g} f_i + \eta, \qquad (19)$$

where M is the number of degrees of freedom, d is the number of common order, $d = 6 - \chi$ (χ is a common constraint), f_i is the number of degrees of freedom of the ith joint, g is the number of joints, n is the number of bodies in the mechanism, and η is the number of over-constraints. The screw theory is used to analyze the number of over-constraints η, and the screws in $limb_{A_1B_1C_1}$ are shown in Figure 6.

$$\$_1 = \begin{cases} \hat{\$}_{11} : (a_{11} \quad b_{11} \quad c_{11}; 0 \quad 0 \quad 0 \ ), \\ \hat{\$}_{12} : (a_{21} \quad b_{21} \quad c_{21} \ ; 0 \quad 0 \quad 0), \\ \hat{\$}_{13} : (1 \quad 0 \quad 0 \ ; 0 \quad z_1 \quad -y_1), \\ \hat{\$}_{14} : (0 \quad 1 \quad 0 \ ; -z_1 \quad 0 \quad x_1), \\ \hat{\$}_{15} : (0 \quad 0 \quad 1; y_1 \quad -x_1 \quad 0). \end{cases} \qquad (20)$$

$\$_1^r$ is defined as the reciprocal of screw $\$_1$ such that

$$\$_1^r \circ \$_1 = 0. \qquad (21)$$

The reciprocal screw $\$_1^r$ of $limb_{A_1B_1C_1}$ is derived using Eqs. (20) and (21):

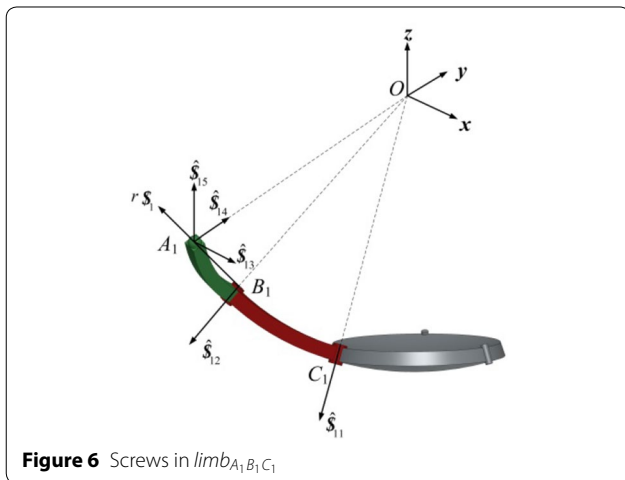$$\$_1^r : \left(\frac{x_1}{z_1} \quad \frac{y_1}{z_1} \quad 1; 0 \quad 0 \quad 0 \right), \qquad (22)$$

Figure 6 Screws in $limb_{A_1B_1C_1}$

where

$$\begin{cases} i_1 = (x_1 \quad y_1 \quad z_1), \ i'_1 = (x'_1 \quad y'_1 \quad z'_1); \\ i_2 = (x_2 \quad y_2 \quad z_2), \ i'_2 = (x'_2 \quad y'_2 \quad z'_2); \\ i_3 = (x_3 \quad y_3 \quad z_3), \ i'_3 = (x'_3 \quad y'_3 \quad z'_3). \end{cases} \qquad (23)$$

$$\begin{cases} i_1 = (x_1 \quad y_1 \quad z_1) = ({}^oA_{11} \quad {}^oA_{12} \quad {}^oA_{13}), \\ j_1 = (a_{21} \quad b_{21} \quad c_{21}) = ({}^oB_{11} \quad {}^oB_{12} \quad {}^oB_{13}), \\ k_1 = (a_{11} \quad b_{11} \quad c_{11}) = ({}^oC_{11} \quad {}^oC_{12} \quad {}^oC_{13}). \end{cases} \qquad (24)$$

The same method is used to obtain the reciprocal screws. $\$_2^r, \$_3^r, \$_1^{r\prime}, \$_2^{r\prime}, \$_3^{r\prime}$ of $limb_{A_2B_2C_2}$, $limb_{A_3B_3C_3}$, $limb_{A'_1B'_1C'_1}$, $limb_{A'_2B'_2C'_2}$, $limb_{A'_3B'_3C'_3}$, $limb_{A_3B_3C_3}$, $limb_{A'_1B'_1C'_1}$, $limb_{A'_2B'_2C'_2}$, $limb_{A'_3B'_3C'_3}$ as follows:

$$\$^r = \begin{cases} \$_1^r : \left(\frac{x_1}{z_1} \quad \frac{y_1}{z_1} \quad 1 ; 0 \ 0 \ 0 \right), \\ \$_2^r : \left(\frac{x_2}{z_2} \quad \frac{y_2}{z_2} \quad 1 ; 0 \ 0 \ 0 \right), \\ \$_3^r : \left(\frac{x_3}{z_3} \quad \frac{y_3}{z_3} \quad 1 ; 0 \ 0 \ 0 \right), \\ \$_1^{r\prime} : \left(\frac{x_{1\prime}}{z_{1\prime}} \quad \frac{y_{1\prime}}{z_{1\prime}} \quad 1 ; 0 \ 0 \ 0 \right), \\ \$_2^{r\prime} : \left(\frac{x_{2\prime}}{z_{2\prime}} \quad \frac{y_{2\prime}}{z_{2\prime}} \quad 1 ; 0 \ 0 \ 0 \right), \\ \$_3^{r\prime} : \left(\frac{x_{3\prime}}{z_{3\prime}} \quad \frac{y_{3\prime}}{z_{3\prime}} \quad 1 ; 0 \ 0 \ 0 \right). \end{cases} \qquad (25)$$

Rank($\$^r$) = 3, so the number of over-constraints $\eta = 3$, $\$_1^r, \$_2^r, \$_3^r, \$_1^{r\prime}, \$_2^{r\prime}, \$_3^{r\prime}$ are nonlinear relative to each other, and $\chi = 0$, and $n = 14, g = 18, d = 6, \sum_{i=1}^{g} f_i = 30$; thus

$$M = 6(14 - 18 - 1) + 30 + 3 = 3. \qquad (26)$$

Therefore, the mechanism has three degrees of freedom, and the cabin's spin becomes a redundant degree of freedom with regard to the w-axis.

3.3 Jacobian Matrix

A Jacobian matrix is a mapping relationship between the angular velocity of active joints and the angular velocity of the moving platform. The establishment of a Jacobian matrix is fundamental to the analysis of forward kinematics in this paper. There are many methods for establishing a Jacobian matrix. In general, a Jacobian matrix is obtained by virtue of derivations of constraining equations. It is troublesome to obtain a Jacobian matrix by virtue of derivations if the constraining equations are too

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 6 of 12

many. For simplicity, in this paper, screw theory is used to build a Jacobian matrix [31–33] for which the moving platform angle velocity $\$_p$ is derived as follows:

$$\$_p = \hat{\$}_{11}\xi_1 + \hat{\$}_{12}\xi_2 + \hat{\$}_{13}\xi_3 + \hat{\$}_{14}\xi_4 + \hat{\$}_{15}\xi_5, \quad (27)$$

where

$$\$_p : (\dot{\lambda} \quad \dot{\varepsilon} \quad \dot{\upsilon}; 0 \quad 0 \quad 0), \quad (28)$$

and where ξ_1, ξ_2, ξ_3, ξ_4, ξ_5 are angular velocities associated with unit screws $\hat{\$}_{11}$, $\hat{\$}_{12}$, $\hat{\$}_{13}$, $\hat{\$}_{14}$, $\hat{\$}_{15}$.

The next step is to establish a relationship between ξ_1 and $\$_p$ while eliminating ξ_2, ξ_3, ξ_4, ξ_5. $^r\$_1$ denotes a reciprocal screw and is derived by means of the following rules:

(1) $^r\$_1$ is through the center of the spherical joint of the A_1-axis and B_1-axis;
(2) $^r\$_1$ does not intersect with the C_1-axis.

Figure 6 shows the position of $^r\$_1$,

$$^r\$_1 = [(\boldsymbol{i} - \boldsymbol{j}); \boldsymbol{i} \times (\boldsymbol{i} - \boldsymbol{k})], \quad (29)$$

and from derivation of the rules, the reciprocal product is

$$^r\$_1 \circ \hat{\$}_{1i} = 0, i = 2, 3, 4, 5. \quad (30)$$

Substituting Eq. (29) into Eq. (27),

$$^r\$_1 \circ \$_p = {}^r\$_1 \circ \hat{\$}_{11}\xi_1. \quad (31)$$

Substituting Eq. (20) and Eq. (26) into Eq. (31) results in

$$[\boldsymbol{i} \times (\boldsymbol{i} - \boldsymbol{j})]\begin{pmatrix} \dot{\lambda} \\ \dot{\varepsilon} \\ \dot{\upsilon} \end{pmatrix} = [\boldsymbol{i} \times (\boldsymbol{i} - \boldsymbol{j}) \cdot \boldsymbol{k}]\xi_1, \quad (32)$$

where $\xi_1 = \dot{\theta}_1$. The next step is to establish the equations of the other limbs of Eq. (26) by the same means.

The following equation is associated with all of the limbs.

$$\boldsymbol{J}_q \dot{\boldsymbol{\omega}} = \boldsymbol{J}_p \dot{\boldsymbol{\theta}}, \quad (33)$$

where

$$\dot{\boldsymbol{\omega}} = (\dot{\lambda} \quad \dot{\varepsilon} \quad \dot{\upsilon})^T, \dot{\boldsymbol{\theta}} = \begin{pmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}'_1 & \dot{\theta}'_2 & \dot{\theta}'_3 \end{pmatrix}^T,$$

$$\boldsymbol{J}_q = \begin{pmatrix} \boldsymbol{i}_1 \times \boldsymbol{j}_1 & \boldsymbol{i}_2 \times \boldsymbol{j}_2 & \boldsymbol{i}_3 \times \boldsymbol{j}_3 & \boldsymbol{i}'_1 \times \boldsymbol{j}'_1 & \boldsymbol{i}'_2 \times \boldsymbol{j}'_2 & \boldsymbol{i}'_3 \times \boldsymbol{j}'_3 \end{pmatrix}^T, \quad (35)$$

$\boldsymbol{J}_q$ is not a square matrix, so herein, the Jacobian matrix is defined as

$$\boldsymbol{J} = \boldsymbol{J}_p^{-1}\boldsymbol{J}_q. \quad (36)$$

Rewriting Eq. (32) yields

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\theta}}. \quad (37)$$

4 Inverse Kinematics

In the equations of inverse kinematics, the posture parameters $(\lambda, \varepsilon, \upsilon)$ of the moving platform are given. The objective is to calculate the input angle θ_i and θ'_1, $i = 1, 2, 3$. In the inverse kinematic analysis, the kinematic equations of the 3-RRS spherical parallel manipulator are the same as those of the 3-RRR spherical parallel manipulator, but the 3-RRR spherical parallel manipulator is an over-constrained mechanism that is difficult to practically assemble. That is why we chose a flight simulator model based on 3-RRS spherical parallel manipulators.

When considering the independently redundant DOF of the cabin, because there are three outputs and eight inputs, it is innumerable about the inverse solutions. If we ignore the independently redundant DOF of the cabin, and fix the independent DOF of cabin, the analysis of the inverse kinematics is as follows.

The angle α_2 between A_i-axis and B_i-axis is invariable, according to GOSSELIN's method [34]

$$\boldsymbol{j}_i \cdot \boldsymbol{i}_i = \cos\alpha_2, i = 1, 2, 3. \quad (38)$$

Substituting Eq. (1) and Eq. (16) into Eq. (38), yields

$$E_i \sin\theta_i + F_i \cos\theta_i = G_i, i = 1, 2, 3, \quad (39)$$

$$\begin{cases} E_i = (\cos\phi_i\sin\alpha_1\boldsymbol{i}_{i1} + \sin\alpha_1\sin\phi_i\boldsymbol{i}_{i2}), \\ F_i = (-\cos\beta_1\sin\phi_i\sin\alpha_1\boldsymbol{i}_{i1} + \cos\beta_1\cos\phi_i\sin\alpha_1\boldsymbol{i}_{i2} \\ \quad - \sin\alpha_1\sin\beta_1\boldsymbol{i}_{i3}), \\ G_i = - (\cos\alpha_1\sin\beta_1\sin\phi_i\boldsymbol{i}_{i1} - \cos\alpha_1\cos\phi_i\sin\beta_1\boldsymbol{i}_{i2} \\ \quad - \cos\alpha_1\cos\beta_1\boldsymbol{i}_{i3} - \cos\alpha_2). \end{cases} \quad (40)$$

$$\boldsymbol{J}_p = \begin{pmatrix} \boldsymbol{i}_1 \times \boldsymbol{j}_1 \cdot \boldsymbol{k}_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{i}_2 \times \boldsymbol{j}_2 \cdot \boldsymbol{k}_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{i}_3 \times \boldsymbol{j}_3 \cdot \boldsymbol{k}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{i}'_1 \times \boldsymbol{j}'_1 \cdot \boldsymbol{k}'_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{i}'_2 \times \boldsymbol{j}'_2 \cdot \boldsymbol{k}'_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{i}'_3 \times \boldsymbol{j}'_3 \cdot \boldsymbol{k}'_3 \end{pmatrix}, \quad (34)$$

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 7 of 12

$$\boldsymbol{i}_i = \begin{pmatrix} {}^{o}\!A_{i1} & {}^{o}\!A_{i2} & {}^{o}\!A_{i3} \end{pmatrix}^{\mathrm{T}}, i = 1, 2, 3. \tag{41}$$

x_i is defined as Eq. (42) as

$$x_i = \tan(\theta_i/2), i = 1, 2, 3. \tag{42}$$

Then Eq. (38) is transformed into Eq. (43)

$$(G_i - F_i)x_i^2 - 2E_i x_i + G_i + F_i = 0, i = 1, 2, 3. \tag{43}$$

Simplifying Eq. (43) yields Eq. (44)

$$L_i x_i^2 + M_i x_i + N_i = 0, i = 1, 2, 3, \tag{44}$$

where $L_i = G_i - F_i, M_i = -2E_i, G_i + F_i = N_i$.
Then

$$x_i = \left(-M_i \pm \sqrt{M_i^2 - 4L_i N_i} \right) / 2L_i, i = 1, 2, 3. \tag{45}$$

Finally

$$\theta_i = 2\arctan(x_i), i = 1, 2, 3. \tag{46}$$

By repeating Eqs. (38)–(45), θ_i' is derived by the same method:

$$\theta_i' = 2\arctan(x_i'), \ i = 1, \ 2, \ 3. \tag{47}$$

All forms of the combination according to Eq. (46) and Eq. (47) are shown in the Table 5 of Appendix 1.

5 Forward Kinematics

For the forward kinematics, without considering the independently redundant DOF of the cabin, the angles of the input θ_i and $\theta_i', i = 1, 2, 3$ are given, and the posture $(\lambda, \varepsilon, \upsilon)$ of the moving platform is determined. However, considering the redundant actuation, the DOF of the moving platform of the flight simulator is three; thus, it can only choose three angles of the input at one time. Herein, we choose $\theta_i(0 < \theta_i < \pi, i = 1, 2, 3.)$ as the inputs, and the remaining three angles of the input $\theta_i'(0 < \theta_i' < \pi, i = 1, 2, 3)$ are obtained via inverse kinematics. In general, the forward kinematics of a parallel manipulator is fairly complicated when using the analytical method, and many solutions are derived. Moreover, in practical application, the configuration of a manipulator is just one of the forward solutions. The motion of the flight simulator is continuous and the control system is based on a dynamical model of MIMO (multiple input multiple out) that needs to obtain the posture of the moving platform in real-time using the forward equations. Thus, an effective numerical method is employed to solve the forward equations. The numerical method was first used in Stewart's parallel manipulator [35].

The basic idea is to assume initial posture angles $\boldsymbol{\omega}_0 = (\lambda_0, \varepsilon_0, \upsilon_0)^{\mathrm{T}}$ of the moving platform, and the

corresponding input angles $\boldsymbol{p}_0 = \left(\theta_1^0, \theta_2^0, \theta_3^0 \right)^{\mathrm{T}}$ are obtained according to the initial posture $\boldsymbol{\omega}_0 = (\lambda_0, \varepsilon_0, \upsilon_0)^{\mathrm{T}}$ by inverse kinematics. At the same time, giving three initial input angles chosen from the six input angles, assuming that $'\boldsymbol{p}_0 = \left('\theta_1, '\theta_2, '\theta_3 \right)^{\mathrm{T}}$ then

$$\Delta \boldsymbol{p}^{(1)} = '\boldsymbol{p}_0 - \boldsymbol{p}_0, \tag{48}$$

defining

$$\begin{cases} \Delta \boldsymbol{p}^{(1)} = \left(\Delta\theta_1^{(1)}, \ \Delta\theta_2^{(1)}, \ \Delta\theta_3^{(1)} \right), \\ \Delta \boldsymbol{\omega}^{(1)} = \left(\Delta\lambda_1^{(1)}, \ \Delta\varepsilon_2^{(1)}, \ \Delta\upsilon_3^{(1)} \right). \end{cases} \tag{49}$$

Then

$$\Delta \boldsymbol{\omega}^{(1)} = \boldsymbol{J}_{\mathrm{s}} \Delta \boldsymbol{p}^{(1)}, \tag{50}$$

The initial posture becomes

$$\boldsymbol{\omega}_1 = \boldsymbol{\omega}_0 + \Delta\boldsymbol{\omega}^{(1)} = \left(\lambda_0 + \Delta\lambda^{(1)}, \varepsilon_0 + \Delta\varepsilon^{(1)}, \upsilon_0 + \Delta\upsilon^{(1)} \right), \tag{51}$$

$\boldsymbol{J}_{\mathrm{s}}$ is 3×3 dimensional sub Jacobian matrix of $\boldsymbol{J}$, where

$$\boldsymbol{J}_{\mathrm{s}} = \begin{pmatrix} \boldsymbol{i}_1 \times \boldsymbol{j}_1 \cdot \boldsymbol{k}_1 & 0 & 0 \\ 0 & \boldsymbol{i}_2 \times \boldsymbol{j}_2 \cdot \boldsymbol{k}_2 & 0 \\ 0 & 0 & \boldsymbol{i}_3 \times \boldsymbol{j}_3 \cdot \boldsymbol{k}_3 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{i}_1 \times \boldsymbol{j}_1 \\ \boldsymbol{i}_2 \times \boldsymbol{j}_2 \\ \boldsymbol{i}_3 \times \boldsymbol{j}_3 \end{pmatrix}. \tag{52}$$

The above process is repeated, and the iterative equation is

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \boldsymbol{J}_{\mathrm{s}}^{-1}(\boldsymbol{\omega}_k)(\boldsymbol{p'}_0 - \boldsymbol{p}_k). \tag{53}$$

The iteration is stopped by the constraint condition $max\left| \Delta\theta_i^k \right| < \delta, i = 1, 2, 3.$ $\Delta\theta_i^k$ is an element of

$\Delta\boldsymbol{p}^{(k)}$, and δ is the permissible error. $\boldsymbol{J}_{\mathrm{s}}$ must be nonsingular when the manipulator performs a continuous motion and the iteration must converge to a unique solution. Eq. (53) is similar to the Newton-Raphson iteration, but the physical and geometrical meanings are more significant, and the whole process of the algorithm is different. Appendix 2 shows a flow chart of the algorithm.

Table 1 Parameters of the manipulator

α_1 (rad)	α_2 (rad)	β_1 (rad)	β_2 (rad)
π/5	π/5	π/20	29π/36

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 8 of 12

Figure 7 Position of θ_i
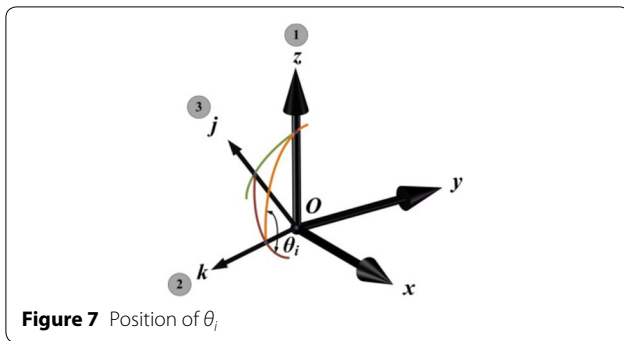
6 Numerical Examples

The parameters of the manipulator are set in Table 1. For initial values $\boldsymbol{\omega}_0 = (0,0,0)^{\mathrm{T}}$ and $\boldsymbol{\omega}_0 = (0.1,0.1,0.1)^{\mathrm{T}}$, the inverse kinematical solutions are $\theta_1^0 = 1.89417$, $\theta_2^0 = 1.89417$, $\theta_3^0 = 1.89417$, and $\theta_1^0 = 1.74548$, $\theta_2^0 = 2.29808$, $\theta_3^0 = 2.05784$ according to the first row of Table 1. Because the spherical analytical method is employed, θ_i^0 is not zero when $\boldsymbol{\omega}_0 = (0,0,0)^{\mathrm{T}}$. Figure 7 contains a schematic chart showing the position of θ_i.

For initial values $'\boldsymbol{p}_0 = (1.89417, 1.89417, 1.89417)^{\mathrm{T}}$ and $\boldsymbol{p}_0 = (1.74548, 2.29808, 2.05784)^{\mathrm{T}}$ along with $\boldsymbol{\omega}_0 = (0.2, 0.2, 0.2)^{\mathrm{T}}$ and $\boldsymbol{\omega}_0 = (-0.025, -0.025, -0.025)^{\mathrm{T}}$ then the forward solutions converge to $\boldsymbol{\omega} = (0,0,0)^{\mathrm{T}}$ and $\boldsymbol{\omega} = (0.1, 0.1, 0.1)^{\mathrm{T}}$, thus proving that the algorithm is correct. Herein, we must point out that as long as $\boldsymbol{\omega}_0$ is less than the range of the workspace, the result must converge to the correct solution, and the solution must be unique. However, the algorithm is like the Newton-Rapshon algorithm in that it is sensitive to the initial values. The results listed in Tables 2 and 3 show that the algorithm is correct. For θ_i', $i = 1, 2, 3$, obtained via the inverse equations, the posture of $\boldsymbol{\omega}_0 = (0.1, 0.1, 0.1)^{\mathrm{T}}$ is shown in Figure 8 when the inverse solution is determined by the first row of Table 1 (identical to column 1 of Table 3).

As shown in Tables 2 and 3, the results are obtained numerically by the forward algorithm and analytically by the inverse equations. The time costs of the numerical forward algorithm are less than 3 ms using MATLAB software by Intel Core i7-3.2G CPU, thus proving that the algorithm of the iteration meets the typical real-time control requirement of less than 6 ms.

To consider the independent DOF of cabin, the forward solution is obtained in two steps. The first step is to obtain the posture $(\lambda, \varepsilon, \upsilon)^{\mathrm{T}}$ by the numerical method.

Table 2 Forward kinematics example 1

Parameters	1	2	3	4	5	6	7	8
$'\theta_1$ (rad)	1.89417	–	–	–	–	–	–	–
$'\theta_2$ (rad)	1.89417	–	–	–	–	–	–	–
$'\theta_3$ (rad)	1.89417	–	–	–	–	–	–	–
λ (rad)	0.00000	–	–	–	–	–	–	–
ε (rad)	0.00000	–	–	–	–	–	–	–
υ (rad)	0.00000	–	–	–	–	–	–	–
θ_1' (rad)	1.89417	1.89417	1.89417	− 1.89417	− 1.89417	− 1.89417	− 1.89417	1.89417
θ_2' (rad)	1.89417	1.89417	− 1.89417	− 1.89417	− 1.89417	1.89417	1.89417	− 1.89417
θ_3' (rad)	1.89417	− 1.89417	− 1.89417	− 1.89417	1.89417	1.89417	− 1.89417	1.89417

Table 3 Forward kinematics example 2

Parameters	1	2	3	4	5	6	7	8
$'\theta_1$ (rad)	1.74548	–	–	–	–	–	–	–
$'\theta_2$ (rad)	2.29808	–	–	–	–	–	–	–
$'\theta_3$ (rad)	2.05784	–	–	–	–	–	–	–
λ (rad)	0.10000	–	–	–	–	–	–	–
ε (rad)	0.10000	–	–	–	–	–	–	–
υ (rad)	0.10000	–	–	–	–	–	–	–
θ_1' (rad)	2.24689	2.24689	2.24689	− 1.72881	− 1.72881	− 1.72881	− 1.72881	2.24689
θ_2' (rad)	1.81504	1.81504	− 2.02941	− 2.02941	− 2.02941	1.81504	1.81504	− 2.02941
θ_3' (rad)	2.08712	− 1.51476	− 1.51476	− 1.51476	2.08712	2.08712	− 1.51476	2.08712

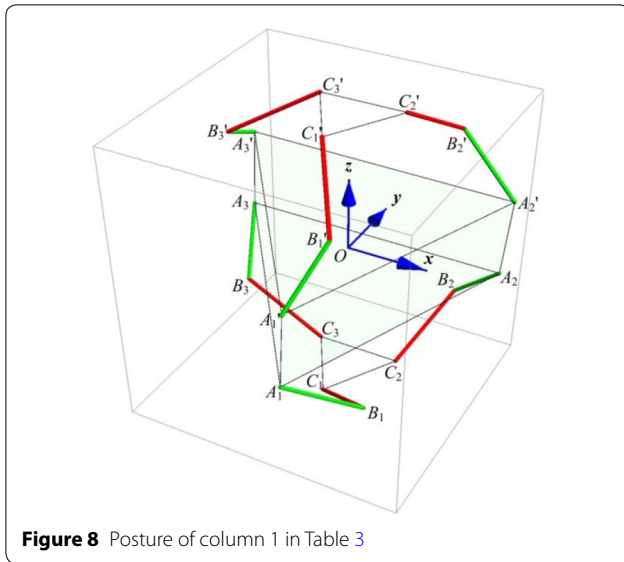Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 9 of 12

Figure 8 Posture of column 1 in Table 3

Then the posture $(\lambda, \varepsilon, \upsilon)^{\mathrm{T}}$ is changed by the cabin's spin with reference to the $\boldsymbol{w}$-axis of the moving coordinate.

Defining ϕ as the angle of the cabin's spin with reference to the $\boldsymbol{w}$-axis, the rotation matrix with reference to the $\boldsymbol{w}$-axis of the moving coordinate is

$$\boldsymbol{R}(\boldsymbol{w}, \phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{54}$$

Defining $(\lambda', \varepsilon', \upsilon')^{\mathrm{T}}$ as the final posture by virtue of the spin of cabin, then

$$(\lambda', \varepsilon', \upsilon')^{\mathrm{T}} = \boldsymbol{R}(\boldsymbol{w}, \phi)(\lambda, \varepsilon, \upsilon)^{\mathrm{T}}, \tag{55}$$

the results of the numerical example are shown in Table 4.

7 Conclusions

(1) The flight simulator for a fighter-aircraft with a hybrid configuration of symmetrically double parallel manipulators named Twins is presented.

(2) Twins is a multi-functional flight simulator. It can be used as a normal flight simulator like Stewart mechanism. And when the cabin spins that will be a flight simulator of fighter-aircraft.

(3) Screw theory is used to establish the Jacobian matrix that simplifies the process of establishing the Jacobian by the means of closed equations.

(4) A numerically forward kinematics is adopted by the inverse kinematics and Jacobian matrix and the method is more simple.

Authors' Contributions
Y-FF was in charge of the whole trial; C-CZ wrote the manuscript; C-CZ assisted with sampling and laboratory analyses. All authors have read and approved the final manuscript.

Authors' Information
Chang-Chun Zhou, born in 1981, is currently a PhD candidate at *Robot Research Center, Beijing Jiaotong University, China*. His research interests include robotics and dynamics.

Yue-Fa Fang, born in 1958, is currently a professor at *Beijing Jiaotong University, China*. He received his PhD degree from *Staffordshire University, UK*, in 1994. His research interest is parallel manipulator.

Appendix 1
See Table 5.

Table 4 Forward kinematics example with cabin' spin

θ'_1 (rad)	θ'_2 (rad)	θ'_3 (rad)	λ (rad)	ε (rad)	υ (rad)	φ (rad)	λ' (rad)	ε' (rad)	υ' (rad)
1.74548	2.29808	2.05784	0.10000	0.10000	0.10000	$3\pi/5$	-0.12601	0.06420	0.10000

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 10 of 12

Table 5 Solution sets of inverse kinematics

Parameters	θ_1	θ_2	θ_3	θ_1'	θ_2'	θ_3'
1	+	+	+	+	+	+
2	+	+	+	+	+	−
3	+	+	+	+	−	−
4	+	+	+	−	−	−
5	+	+	+	−	−	+
6	+	+	+	−	+	+
7	+	+	+	−	+	−
8	+	+	+	+	−	+
9	+	+	−	+	+	+
10	+	+	−	+	+	−
11	+	+	−	+	−	−
12	+	+	−	−	−	−
13	+	+	−	−	−	+
14	+	+	−	−	+	+
15	+	+	−	−	+	−
16	+	+	−	+	−	+
17	+	−	−	+	+	+
18	+	−	−	+	+	−
19	+	−	−	+	−	−
20	+	−	−	−	−	−
21	+	−	−	−	−	+
22	+	−	−	−	+	+
23	+	−	−	−	+	−
24	+	−	−	+	−	+
25	−	−	−	+	+	+
26	−	−	−	+	+	−
27	−	−	−	+	−	−
28	−	−	−	−	−	−
29	−	−	−	−	−	+
30	−	−	−	−	+	+
31	−	−	−	−	+	−
32	−	−	−	+	−	+
33	−	−	+	+	+	+
34	−	−	+	+	+	−
35	−	−	+	+	−	−
36	−	−	+	−	−	−
37	−	−	+	−	−	+
38	−	−	+	−	+	+
39	−	−	+	−	+	−
40	−	−	+	+	−	+
41	−	+	+	+	+	+
42	−	+	+	+	+	−
43	−	+	+	+	−	−
44	−	+	+	−	−	−
45	−	+	+	−	−	+
46	−	+	+	−	+	+
47	−	+	+	−	+	−
48	−	+	+	+	−	+
49	−	+	−	+	+	+

Zhou and Fang *Chin. J. Mech. Eng.* (2018) 31:55

Page 11 of 12

Table 5 (continued)

Parameters	θ_1	θ_2	θ_3	θ_1'	θ_2'	θ_3'
50	−	+	−	+	+	−
51	−	+	−	+	−	−
52	−	+	−	−	−	−
53	−	+	−	−	−	+
54	−	+	−	−	+	+
55	−	+	−	−	+	−
56	−	+	−	+	−	+
57	+	−	+	+	+	+
58	+	−	+	+	+	−
59	+	−	+	+	−	−
60	+	−	+	−	−	−
61	+	−	+	−	−	+
62	+	−	+	−	+	+
63	+	−	+	−	+	−
64	+	−	+	+	−	+

Appendix 2
Numerical algorithm of the forward kinematics

begin
  clear
  clc
  initial parameters;
  k=1;
  while k **do**
    inversely kinematical equations;
    $\Delta p^k = {}'p_k - p_k;$
    if $max|\Delta\theta_i^{(k)}| < \delta, i = 1, 2, 3$ **then**
    | break;
    else
      Jacobian matrix;
      $\Delta\omega^k = J_s^{-1}\Delta p^k;$
      $\omega_{k+1} = \omega_k + \Delta\omega^k;$
      $k = k + 1;$
    end
  end
end
Output: results

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References
[1] E N Johnson, S Mishra. Flight simulation for the development of an experimental UAV. *Clays & Caly Minerals*, 2002, 44(6): 825–834.
[2] D S Wu, H Du. Adaptive sliding control of six-DOF flight simulator motion platform. *Chinese Journal of Aeronautics*, 2007, 20(5): 425–433.
[3] D Stewart. A platform with six degrees of freedom. *Proceeding of the Institution of Mechanical Engineers*, 1965, 180(1): 1615–1622.
[4] M Nahon, R Richard, C M Gosselin. A comparison of flight simulator motion-base architectures. *Proceedings of the CEAS Symposium on Simulation Technology*, 2003: 1–16.
[5] A Karger, M Husty. Classification of all self-motions of the original Stewart-Gough platform. *Computer-Aided Design*, 1998, 30(3): 205–215.
[6] B L Aponso, S D Beard, J A Schroeder. The NASA ames vertical motion simulator—a facility engineered for realism. *Proceeding of the Royal Aeronautical Society Spring Flight Simulation Conference*, 2009: 4.
[7] H Gu, D Wu, H Liu. Development of a novel low-cost flight simulator for pilot training. *World Academy of Science, Engineering and Technology*, 2009, 60: 685–689.
[8] J Kim, J C Hwang, S K Jin, et al. Eclipse II: a new parallel mechanism enabling continuous 360-degree spinning plus three-axis translational motions. *IEEE Transactions on Robotics & Automation*, 2002, 18(3): 367–373.
[9] N A Pouliot, C M Gosselin, M A Nahon. Motion simulation capabilities of three-degree-of-freedom flight simulators. *Journal of Aircraft*, 1998, 11(2): 9–17.
[10] H Shui, C Chen, D Oetomo. 7 DOF arm type haptic interface for teleoperation and virtual reality system. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, Canada, October 14–19, 1998: 1266–1231.
[11] K E Neumann. Next generation tricept: a true revolution in parallel kinematics. *Proceedings of the 4th Chemnitz Parallel kinematics Seminar*, Zwickau, Germany, December 12–15, 2004: 591–594.
[12] T Huang, M Li, X Zhao, et al. Conceptual design and dimensional synthesis for a 3-dof module of the trivariant—a novel 5-DOF reconfigurable hybrid robot. *IEEE Transactions on Robotics*, 2005, 21(3): 449–456.
[13] M K Lee, K W Park. Kinematic and dynamic analysis of a double parallel manipulator for enlarging workspace and avoiding singularities. *IEEE Transactions on Robotics and Automation*, 1999, 15(6): 1024–1034.
[14] M K Lee, K W Park. Workspace and singularity analysis of a double parallel manipulator. *IEEE Transactions on Mechatronics,* 2000, 5(4): 367–375.
[15] L W Tsai, S Joshi. Kinematic analysis of 3-DOF position mechanisms for use in hybrid kinematic machines. *Journal of Mechanical Design*, 2004, 124(2): 245–253.

Zhou and Fang *Chin. J. Mech. Eng.* *(2018) 31:55*

Page 12 of 12

[16] Y J Zhao, F Gao, W Li, et al. Development of 6-DOF parallel seismic simulator with novel redundant actuation. *Mechatronics*, 2009, 19(3): 422–427.

[17] M G Mohamed, C M Gosselin. Design and analysis of kinematically redundant parallel manipulators with configurable platforms. *IEEE Transactions on Robotics*, 2005, 21(3): 277– 287.

[18] K E Zanganeh, J Angeles. Instantaneous kinematics and design of a novel redundant parallel manipulator. *Proceedings of the IEEE International Conference on Robotics & Automation*, San Diego, USA, May 8–13, 1994: 3043–3048.

[19] K E Zanganeh, J Angeles. Mobility and position analyses of a novel redundant parallel manipulator. *Proceedings of the IEEE International Conference on Robotics & Automation*, San Diego, USA, May 8–13, 1994: 3049–3054.

[20] J Wang, C M Gosselin. Kinematic analysis of 3-DOF position mechanisms for use in hybrid kinematic machines. *Journal of Design*, 2004, 126(1): 109–118.

[21] S Nokleby, R Fisher, R Podhorodeski, et al. Force capabilities of redundantly-actuated parallel manipulators. *Mechanism and Machine Theory*, 2005, 40(5): 578–599.

[22] J Kim, F C Park, S J Ryu, et al. Design and analysis of a redundantly actuated parallel mechanism for rapid machining. *IEEE Transactions on Robotics and Automation*, 2001, 17(4): 423–434.

[23] J F Li, R Fei, J Fan. Effects of actuator disposition and redundant actuation on performance of the tricept parallel mechanism. *Chinese Journal of Mechanical Engineering*. 2008, 44(1): 31–40.

[24] C M Gosselin, E Lavoie. On the kinematic design of spherical three-degree-of-freedom parallel manipulators. *Journal of Robotics Research*, 1993, 12(4): 394–402.

[25] Y F Fang, L W Tsai. Structure synthesis of a class of 3-DOF rotational parallel manipulators. *IEEE Transactions on Robotics and Automation*, 2004, 20(1): 117–121.

[26] R D Gregorio, The 3-RRS wrist: a new, simple and non-overconstrained spherical parallel manipulator. *Journal of Mechanical Design*, 2004, 126(5): 850–855.

[27] X W Kong, C M Gosselin. Type synthesis of three-degree- of-freedom spherical parallel manipulator. *Journal of Robotics Research*, 2004, 23(3): 237–245.

[28] M Karouia, J M Herve. A family of novel orientational 3-DOF parallel robots. *International Centre for Mechanical Sciences*, 2002: 359–368.

[29] C M Gosselin, J Angeles. The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *Journal of Mechanical Design*, 1989, 111(2): 202–207.

[30] Z Huang, L F Kong, Y F Fang. *The mechanism and control theory of the parallel robot*. Beijing: China Machine Press, 1997. (in Chinese)

[31] J S Dai. *Geometrical foundations and screw algebra for mechanisms and robotics*. Beijing: Higher Education Press, 2014. (in Chinese)

[32] T Zhao, M Geng, Y Chen, et al. Kinematics and dynamics Hessian matrices of manipulators based on screw theory. *Chinese Journal of Mechanical Engineering*, 2015, 28(2): 226–235.

[33] Z Huang, X W Kong, T S Zhao. *Advanced spatial mechanism*. Beijing: Higher Education Press, 2006. (in Chinese)

[34] C M Gosselin, J Angeles. A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, 1991, 113(3): 220–226.

[35] H Jiang, K P Zheng, X CH Wang, et al. Effective method of forward displacement analysis for virtual axis machine tool. *Journal of Xi'an Jiaotong University*, 2002, 36(11): 1185–1189.