Abstract: This paper presents the forward displacement analysis of an 8-PSS (prismatic-spherical-spherical) redundant parallel manipulator whose moving platform is linked to the base platform by eight kinematic chains consisting of a PSS joint and a strut with fixed length. A general approximation algorithm is used to solve the problem. To avoid the extraction of root in the approximation process, the forward displacement analysis of the 8-PSS redundant parallel manipulator is transformed into another equivalent problem on the assumption that the strut is extensible while the slider is fixed. The problem is solved by a modified approximation algorithm which predicates that the manipulator will move along a pose vector to reduce the difference between the desired configuration and an instantaneous one, and the best movement should be with minimum norm and least quadratic sum. The characteristic of this modified algorithm is that its convergence domain is larger than that of the general approximation algorithm. Simulation results show that the modified algorithm is general and can be used for the forward displacement analysis of the redundant parallel manipulator actuated by a revolute joint.

Keywords: redundant parallel manipulator; forward displacement analysis; approximation algorithm

1 Introduction

A parallel manipulator is a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains. Thanks to the base-mounted or near base-mounted actuation arrangement, it is widely recognized that the parallel manipulator has some potential advantages such as high speed, rigidity/weight ratio, accuracy and so on. To fully realize these potential advantages, redundancy is introduced to improve the ability and performance of parallel manipulators [1-10]. It is believed that there are many advantages for the redundant parallel manipulator such as avoiding kinematic singularities, increasing workspace, improving dexterity, and enlarging load capability. There are mainly two different types of redundancy for the parallel manipulator [9-10]: a) kinematic redundancy and b) actuation redundancy. A parallel manipulator is said to be kinematically redundant when its mobility of the mechanism is larger than the required degrees of freedom of the moving platform. On the other hand, a parallel manipulator is called redundantly actuated when the number of actuators is larger than the mobility of the mechanism. Redundant actuation in a parallel manipulator can be implemented by the following approaches: a) to actuate some of the passive joints within the branches of a parallel manipulator; b) to add additional branches...
by beyond the minimum necessary to actuate the parallel manipulator; and c) to combine the foregoing two approaches.

Forward displacement analysis is one of the important issues in the development of parallel manipulator. It is usually used for computer simulation and virtual prototype building. Furthermore, the motion planning and control of the manipulator calls for the forward displacement analysis. Efficient solutions to this problem allow the algorithms for the control and motion planning of the manipulator configuration to be improved. The direct kinematic problem is to obtain the orientation and position of the moving platform with respect to the base plate from the knowledge of the lengths of its limbs and the geometry of the base and moving plate. It involves strongly nonlinear difficult issues and remains yet an incompletely solved problem. Though much attention has been paid to the direct position problem of the general parallel manipulator [11-15] with the numerical method and analytical method, little investigation [16] has been done on the direct position problem of redundant parallel manipulators due to their complex structures.

By taking the 8-PSS redundant parallel manipulator as the research object, in this paper we proposed a numerical algorithm for the forward displacement analysis of the redundant parallel manipulator. We transformed the problem into an equivalent form and used a modified approximation algorithm with an obvious physical significance to solve it. The validity and effectiveness of the algorithm was verified with simulation results. The proposed method is applicable to the forward displacement analysis of a parallel manipulator actuated by a revolute joint.

2 System description

The 8-PSS redundant parallel manipulator is shown in Fig. 1. It consists of a moving platform and eight sliders. In each kinematic chain, the platform and the slider are connected via spherical ball bearing joints by a strut of fixed length. Each slider is driven by an actuated joint via a linear ball screw. The lead screws $B_1$, $B_2$, $B_3$ and $B_4$ are vertical to and $B_5$, $B_6$, $B_7$ and $B_8$ are parallel with the ground. They are orthogonal to each other.

![Fig. 1 Schematic diagram of the 8-PSS redundant parallel manipulator, where $A_{1,8}$ and $C_{1,8}$ are the positions of spherical joint of the strut linked to the moving platform and the slider, $B_{1,8}$ are the positions of the lead screws linked to the base platform; $a$ is half the distance between two vertical sliders parallel to each other; $b$ is the half length of the moving square platform; $c$ is half the distance between two horizontal sliders parallel to each other; $h$ is the thickness of the moving platform; $h_1$ is the horizontal and $h_2$ is the vertical distances from the center point of the square $B_1B_2B_3B_4$ to the point $B_{i,5,6,7,8}$](image-url)
The vector diagram of the PSS kinematic chain is shown in Fig. 2.

For the purpose of analysis, the following coordinate systems are defined. The coordinate system $O-xyz$ is attached to the fixed base and another moving coordinate frame $O'-u'v'w'$ is located at the mass center of the moving platform. The pose of the moving platform can be described with a position vector $r$ from $O$ to $O'$ and a rotation matrix $R$. Let the rotation matrix be defined by the roll, pitch, and yaw angles, namely, a rotation of $\phi_1$ around the fixed $x$ axis, followed by a rotation of $\phi_2$ around the fixed $y$ axis, and a rotation of $\phi_3$ around the fixed $z$ axis. Thus, the rotation matrix is

$$R = \text{Rot}(z, \phi_3) \text{Rot}(y, \phi_2) \text{Rot}(x, \phi_1). \quad (1)$$

3 Direct position analysis

As shown in Fig. 2, the closed-loop position equation associated with the $i$th kinematic chain can be written as

$$r + a_i = l_i w_i + b_i + d_i + q_i e_i, \quad (2)$$

where $q_i$ is the slider position variable; $e_i$ is the unit vector along the lead screw; $w_i$ is the unit vector along the strut $C_i A_i$; $a_i$ is the vector $O A_i$; $b_i$ is the vector $O B_i$; $d_i$ is the vector from the lead screw to the center point of the joint $C_i$ respectively; and $l_i$ is the length of the strut $C_i A_i$.

The derivative of Eq. (2) with respect to time is

$$\dot{q}_i e_i + \omega_i \times l_i w_i = v + \omega \times a_i, \quad (3)$$

where $\omega$ and $v$ denote the angular velocity of the strut $C_i A_i$ and the linear velocity of the moving platform.

Calculating the dot product of both sides of Eq. (3) with $w_i$ yields

$$\dot{q}_i = \left[ \frac{w_i^T}{w_i^T e_i, \left( a_i \times w_i \right)^T, w_i^T e_i} \right] \left[ v, \omega \right]. \quad (4)$$

Rewriting Eq. (4) in a matrix form yields

$$\dot{q} = J^{-1} J \dot{X} = JX, \quad (5)$$

where

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \\ \dot{q}_7 \\ \dot{q}_8 \end{bmatrix}^T, \quad (6)$$

$$X = \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (7)$$

$$J_\dot{q} = \text{diag} \left( w_1 e_1, w_2 e_2, w_3 e_3, w_4 e_4, w_5 e_5, w_6 e_6, w_7 e_7, w_8 e_8 \right). \quad (8)$$

$$J_x = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ a_1 \times w_1 \\ a_2 \times w_2 \\ a_3 \times w_3 \\ a_4 \times w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ a_5 \times w_5 \\ a_6 \times w_6 \\ a_7 \times w_7 \\ a_8 \times w_8 \end{bmatrix}^T, \quad (9)$$

where $J$ is the Jacobian matrix which maps the velocity vector $X$ into the slider velocity vector $\dot{q}$.

3.1 Approximation algorithm 1

Suppose the desired approximation precision is $e$. Let $X_0$ be the initial estimated pose of the moving platform. The direct position problem can be solved as the
The following general approximation algorithm.

Step 1: Calculate the slider position variables of the kinematic chains corresponding to the initial estimated pose \( X_0 \).

The slider position variables \( q_{0i} \) corresponding to the instantaneous configuration determined by \( X_0 \) can be calculated by

\[
q_{0i} = \sqrt{(r_i + a_{0i} - l_i w_{0i} - h_i - d_i)^2} (r_i + a_{0i} - l_i w_{0i} - h_i - d_i),
\]

where

\[
a_{0i} = \mathbf{R}_{uv}^{-1} a_{ui},
\]

\[
\mathbf{R}_{uv} = \mathbf{Rot}(z, X_0(6,1)) \mathbf{Rot}(y, X_0(5,1)) \mathbf{Rot}(x, X_0(4,1)).
\]

Step 2: Calculate the approximation step that the moving platform needs to move.

The Jacobian matrix of the instantaneous configuration is achieved with Eqs. (5) to (9). The difference between the input slider position variables and the instantaneous slider position variables is

\[
\Delta q = [q_1 - q_{01}, q_2 - q_{02}, q_3 - q_{03}, q_4 - q_{04}, q_5 - q_{05}, q_6 - q_{06}, q_7 - q_{07}, q_8 - q_{08}]^T.
\]

Step 3: Calculate the new pose vector of the moving platform \( X_1 \) which represents the new instantaneous configuration.

\[
X_1 = X_0 + \lambda \Delta X,
\]

where \( \lambda \in (0, \gamma) \) is a coefficient, and \( \gamma \) is a number that can be larger than 1.

Step 4: Calculate the new slider position variables \( q_{1i} \) corresponding to the new instantaneous configuration determined by the pose vector \( X_1 \).

\[
t_1 = \sqrt{(r_i + a_{1i} - l_i w_{1i} - h_i - d_i)^2} (r_i + a_{1i} - l_i w_{1i} - h_i - d_i),
\]

To approximate the desired configuration, the pose vector difference which the moving platform should travel is

\[
\Delta X = J^* \Delta q,
\]

where \( J \) is the Jacobian matrix corresponding to the instantaneous configuration of the parallel manipulator; and \( J^* \) is the Moore-Penrose inverse of the Jacobian matrix. The physical significance of Eq.(15) is that among all possible pose vectors which the manipulator will move along to reduce the difference between the desired configuration and an instantaneous one, the best movement should be with minimum norm and least quadratic sum. It is taken as the approximation step.

Step 5: If the following condition

\[
\|q - q_{0i}\| \leq e
\]

is satisfied, stop the approximation. The instantaneous configuration is the desired one. Herein \( || \cdot || \) is the norm of the vector which can be 1–norm, 2–norm and \( \infty \)–norm. In this paper, it is selected to be 2–norm.

If the above condition is not satisfied, let

\[
X_0 = X_1.
\]

Go to Step 2 to continue the approximation.

It must be noted that the approximation algorithm is
Due to Eq. (13), the result of the slider position variable is often a complex number. There is no physical significance for a complex slider position variable. Therefore, the approximation procedure often has to be interrupted or finishes with a result of no physical significance. To avoid the extraction of root in the approximation process, the direct position problem of the 8-PSS redundant parallel manipulator is transformed into another equivalent form.

3.2 Approximation algorithm 2

Supposing that the strut $C_iA_i$ is extensible while the slider is fixed, the main idea of the solution two is to determine the position of the point $A_i$ when the length of the struts is given.

$$\dot{l}_w + \omega \times l_w = v + \omega \times a_i.$$  \hfill (23)

Calculating the dot product of both sides of Eq. (23) with $w_i$ yields

$$\dot{l}_i = \left[ w_i^T \ (a_i \times w_i)^T \right] \left[ \begin{array}{c} v \ \omega \end{array} \right].$$ \hfill (24)

Rewriting Eq. (24) in the matrix form yields

$$\dot{L} = J \dot{X},$$ \hfill (25)

where

$$\dot{L} = \left[ \begin{array}{cccc} \dot{l}_1 \\ \dot{l}_2 \\ \dot{l}_3 \\ \dot{l}_4 \\ \dot{l}_5 \\ \dot{l}_6 \\ \dot{l}_7 \\ \dot{l}_8 \end{array} \right],$$ \hfill (26)

$$\dot{X} = \left[ \begin{array}{c} v \\ \omega \end{array} \right].$$ \hfill (27)

The equivalent direct kinematics problem is solved as the following.

Step 1: Calculate the strut length variables of the kinematic chains corresponding to the initial estimated pose $X_0$.

The initial length of the supposed extensible link corresponding to the instantaneous configuration determined by the $X_0$ can be calculated by

$$l_{0i} = \sqrt{(r_0 + a_{i0} - b_i - d_i - qe_i)^2 + (r_0 + a_{i0} - b_i - d_i - qe_i)^2},$$ \hfill (29)

where

$$a_{i0} = \mathbf{R}_{v0} \mathbf{a}_{i0},$$ \hfill (30)

$$\mathbf{R}_{v0} = \text{Rot}(z, X_v(6,1))\text{Rot}(y, X_v(5,1))\text{Rot}(x, X_v(4,1)).$$ \hfill (31)

Step 2: Calculate the approximation step that the moving platform needs to pass over.

The difference between the input legs length and the supposed instantaneous legs length is

$$\Delta L = \left[ egin{array}{cccc} \dot{l}_1 - l_{01} \\ \dot{l}_2 - l_{02} \\ \dot{l}_3 - l_{03} \\ \dot{l}_4 - l_{04} \\ \dot{l}_5 - l_{05} \\ \dot{l}_6 - l_{06} \\ \dot{l}_7 - l_{07} \\ \dot{l}_8 - l_{08} \end{array} \right]^T.$$ \hfill (32)

To approximate the desired configuration, the pose vector difference which the moving platform should move through is

$$\Delta X = J \Delta L.$$ \hfill (33)

Step 3: Calculate the new pose vector of the moving platform $X_1$ which represents the new instantaneous configuration.

$$X_1 = X_0 + \lambda \Delta X,$$ \hfill (34)

where $\lambda \in (0, \gamma)$ is a coefficient, and $\gamma$ is a number that can be larger than 1.

Step 4: Calculate the new length of the supposed link corresponding to the new instantaneous configuration determined by the pose vector $X_1$.

$$l_{1i} = \sqrt{(r_i + a_{i1} - b_i - d_i - qe_i)^2 + (r_i + a_{i1} - b_i - d_i - qe_i)^2},$$ \hfill (35)

$$L_1 = \left[ \begin{array}{cccc} l_{11} \\ l_{12} \\ l_{13} \\ l_{14} \\ l_{15} \\ l_{16} \\ l_{17} \\ l_{18} \end{array} \right]^T,$$ \hfill (36)

where

$$a_{i1} = \mathbf{R}_{v1} \mathbf{a}_{i1},$$ \hfill (37)

$$\mathbf{R}_{v1} = \text{Rot}(z, \Delta X(6,1))\text{Rot}(y, \Delta X(5,1))\text{Rot}(x, \Delta X(4,1)) \mathbf{R}_{v0}.$$ \hfill (38)

Step 5: If the following condition

$$\|L - L_0\| \leq \epsilon$$ \hfill (39)
is satisfied, stop the approximation. The instantaneous configuration is the desired one. There is no extraction of square root of quadratic equation like Eq. (13) in this approximation. So its convergence domain is larger than that of the approximation algorithm 1. The applicability of the modified approximation algorithm 2 for forward kinematics of the redundant parallel manipulator was verified by simulation tests.

4 Simulation

The forward displacement problem of the 8-PSS redundant parallel manipulator was solved by the presented method, with the structure parameters and the input slider position variables listed in Tables 1 and 2, respectively. The distance from the lead screw to the center point of the joint was \( d = 0.244 \) m. The approximation precision was \( e = 10^{-6} \) m and the coefficient was \( \lambda = 1 \).

The simulation results are shown in Tables 3 to 5, where \( L_i \) is the length vector of the strut corresponding to the instantaneous configuration and \( q_i \) is the slider position variable vector corresponding to the instantaneous configuration in the approximation process.

### Table 1 Parameters of the base platform \( x_{Bi}, y_{Bi} \) and \( z_{Bi} \) measured in the coordinate system \( O-xyz \) and of the moving platform \( x_{Ai}, y_{Ai} \) and \( z_{Ai} \) measured in the coordinate system \( O'-u'v'w'(i=1,2,…,8) \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_{Bi}/m )</th>
<th>( y_{Bi}/m )</th>
<th>( z_{Bi}/m )</th>
<th>( x_{Ai}/m )</th>
<th>( y_{Ai}/m )</th>
<th>( z_{Ai}/m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.400 000</td>
<td>-0.400 000</td>
<td>0.000 000</td>
<td>0.400 000</td>
<td>-0.400 000</td>
<td>-0.166 000</td>
</tr>
<tr>
<td>2</td>
<td>0.400 000</td>
<td>0.400 000</td>
<td>0.000 000</td>
<td>0.400 000</td>
<td>0.400 000</td>
<td>-0.166 000</td>
</tr>
<tr>
<td>3</td>
<td>-0.400 000</td>
<td>0.400 000</td>
<td>0.000 000</td>
<td>-0.400 000</td>
<td>0.400 000</td>
<td>-0.166 000</td>
</tr>
<tr>
<td>4</td>
<td>-0.400 000</td>
<td>-0.400 000</td>
<td>0.000 000</td>
<td>-0.400 000</td>
<td>-0.400 000</td>
<td>-0.166 000</td>
</tr>
<tr>
<td>5</td>
<td>0.400 000</td>
<td>-2.000 000</td>
<td>1.500 000</td>
<td>0.400 000</td>
<td>-0.681 000</td>
<td>-0.037 500</td>
</tr>
<tr>
<td>6</td>
<td>-0.400 000</td>
<td>-2.000 000</td>
<td>1.500 000</td>
<td>-0.400 000</td>
<td>-0.681 000</td>
<td>-0.037 500</td>
</tr>
<tr>
<td>7</td>
<td>-2.000 000</td>
<td>-0.400 000</td>
<td>1.500 000</td>
<td>-0.681 000</td>
<td>-0.400 000</td>
<td>-0.037 500</td>
</tr>
<tr>
<td>8</td>
<td>-2.000 000</td>
<td>0.400 000</td>
<td>1.500 000</td>
<td>-0.681 000</td>
<td>0.400 000</td>
<td>-0.037 500</td>
</tr>
</tbody>
</table>

### Table 2 Input data of the strut length \( l_i \), input slider position variable \( q_i \), and the position parameters of the spherical ball bearing joint \( x_{Ci}, y_{Ci} \) and \( z_{Ci} \) measure in the coordinate system \( O-xyz \) for simulation test \((i=1,2,…,8)\)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( l_i/m )</th>
<th>( q_i/m )</th>
<th>( x_{Ci}/m )</th>
<th>( y_{Ci}/m )</th>
<th>( z_{Ci}/m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000 000</td>
<td>0.203 783</td>
<td>0.400 000</td>
<td>-0.156 000</td>
<td>0.269 783</td>
</tr>
<tr>
<td>2</td>
<td>1.000 000</td>
<td>0.291 963</td>
<td>0.400 000</td>
<td>0.644 000</td>
<td>0.357 963</td>
</tr>
<tr>
<td>3</td>
<td>1.000 000</td>
<td>0.362 339</td>
<td>-0.400 000</td>
<td>0.644 000</td>
<td>0.428 339</td>
</tr>
<tr>
<td>4</td>
<td>1.000 000</td>
<td>0.274 241</td>
<td>-0.400 000</td>
<td>-0.156 000</td>
<td>0.340 241</td>
</tr>
<tr>
<td>5</td>
<td>1.000 000</td>
<td>0.469 648</td>
<td>0.400 000</td>
<td>-1.530 352</td>
<td>1.744 000</td>
</tr>
<tr>
<td>6</td>
<td>1.000 000</td>
<td>0.519 544</td>
<td>-0.400 000</td>
<td>-1.480 456</td>
<td>1.744 000</td>
</tr>
<tr>
<td>7</td>
<td>1.000 000</td>
<td>0.427 421</td>
<td>-1.572 579</td>
<td>-0.400 000</td>
<td>1.744 000</td>
</tr>
<tr>
<td>8</td>
<td>1.000 000</td>
<td>0.495 629</td>
<td>-1.504 371</td>
<td>0.400 000</td>
<td>1.744 000</td>
</tr>
</tbody>
</table>
Tables 3 and 4 show the approximation process with the initial $X_0$ out of the workspace by the approximation algorithm 2 to solve the direct position problem of the 8-PSS redundant parallel manipulator. However, when using the approximation algorithm 1, with the initial $X_0$ out of the workspace of the 8-PSS redundant parallel manipulator, interruption occurs at the first step due to Eq. (13) (Tables 5 and 6). The convergence domain of the approximation algorithm 2 is much larger than that of the approximation algorithm 1. The number of the approximation step in Tables 3 and 4 is only 4, suggesting that the approximation algorithm 2 is effective.

Table 3  Approximation process by approximation algorithm 2, where $X$ is the estimated configuration, $X_0$ is the initial, and $X_i$ ($i=1,2,...,6$) is the $i$th calculated instantaneous configurations of the moving platform; $\phi_x$, $\phi_y$ and $\phi_z$ are a rotation of around the fixed $x$, $y$ and $z$ axes, respectively

<table>
<thead>
<tr>
<th>$X$</th>
<th>$x$/m</th>
<th>$y$/m</th>
<th>$z$/m</th>
<th>$\phi_x$/rad</th>
<th>$\phi_y$/rad</th>
<th>$\phi_z$/rad</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_0$</td>
<td>2.000 000</td>
<td>0.000 000</td>
<td>1.860 000</td>
<td>0.000 000</td>
<td>0.000 000</td>
<td>0.000 000</td>
</tr>
<tr>
<td>$X_1$</td>
<td>0.162 524</td>
<td>1.621 023</td>
<td>2.505 516</td>
<td>0.076 108</td>
<td>0.212 021</td>
<td>−0.102 147</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.864 141</td>
<td>0.219 572</td>
<td>1.693 613</td>
<td>0.057 158</td>
<td>0.035 962</td>
<td>−0.066 887</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.081 816</td>
<td>0.405 234</td>
<td>1.703 225</td>
<td>0.097 120</td>
<td>0.092 912</td>
<td>−0.078 174</td>
</tr>
<tr>
<td>$X_4$</td>
<td>0.152 144</td>
<td>0.132 888</td>
<td>1.531 071</td>
<td>0.099 244</td>
<td>0.089 674</td>
<td>−0.092 814</td>
</tr>
<tr>
<td>$X_5$</td>
<td>0.100 648</td>
<td>0.103 367</td>
<td>1.502 146</td>
<td>0.099 571</td>
<td>0.099 311</td>
<td>−0.099 213</td>
</tr>
<tr>
<td>$X_6$</td>
<td>0.100 006</td>
<td>0.100 001</td>
<td>1.500 008</td>
<td>0.100 000</td>
<td>0.099 996</td>
<td>−0.099 996</td>
</tr>
<tr>
<td>$X$</td>
<td>0.100 000</td>
<td>0.100 000</td>
<td>1.500 000</td>
<td>0.100 000</td>
<td>0.100 000</td>
<td>−0.100 000</td>
</tr>
</tbody>
</table>

Table 4  Approximation process of leg length $L$ corresponding to Table 3 by approximation algorithm 2, where $L_0$ is the initial input leg length, $L_1, L_2, ..., L_6$ are the $i$th calculated instantaneous leg length, and $l_0, l_1, l_2, ..., l_8$ are the lengths of the $j$th strut corresponding to the instantaneous configuration

<table>
<thead>
<tr>
<th>$L$</th>
<th>$l_0$/m</th>
<th>$l_1$/m</th>
<th>$l_2$/m</th>
<th>$l_3$/m</th>
<th>$l_4$/m</th>
<th>$l_5$/m</th>
<th>$l_6$/m</th>
<th>$l_7$/m</th>
<th>$l_8$/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_0$</td>
<td>2.402 284</td>
<td>2.355 425</td>
<td>2.319 754</td>
<td>2.364 653</td>
<td>2.173 202</td>
<td>2.154 190</td>
<td>2.891 822</td>
<td>2.823 620</td>
<td></td>
</tr>
<tr>
<td>$L_1$</td>
<td>2.385 069</td>
<td>2.361 778</td>
<td>2.489 047</td>
<td>2.511 634</td>
<td>2.511 912</td>
<td>2.585 858</td>
<td>2.148 924</td>
<td>2.179 025</td>
<td></td>
</tr>
<tr>
<td>$L_2$</td>
<td>1.477 013</td>
<td>1.474 523</td>
<td>1.442 300</td>
<td>1.443 585</td>
<td>1.334 386</td>
<td>1.335 847</td>
<td>1.751 875</td>
<td>1.736 706</td>
<td></td>
</tr>
<tr>
<td>$L_3$</td>
<td>1.203 070</td>
<td>1.195 153</td>
<td>1.208 508</td>
<td>1.216 303</td>
<td>1.246 431</td>
<td>1.250 201</td>
<td>1.050 787</td>
<td>1.047 752</td>
<td></td>
</tr>
<tr>
<td>$L_4$</td>
<td>1.033 642</td>
<td>1.036 174</td>
<td>1.031 613</td>
<td>1.029 053</td>
<td>1.022 182</td>
<td>1.023 148</td>
<td>1.051 547</td>
<td>1.047 987</td>
<td></td>
</tr>
<tr>
<td>$L_5$</td>
<td>1.002 016</td>
<td>1.001 629</td>
<td>1.001 438</td>
<td>1.001 826</td>
<td>1.002 263</td>
<td>1.002 170</td>
<td>1.000 905</td>
<td>1.000 433</td>
<td></td>
</tr>
<tr>
<td>$L_6$</td>
<td>1.000 007</td>
<td>1.000 008</td>
<td>1.000 005</td>
<td>1.000 006</td>
<td>1.000 007</td>
<td>1.000 007</td>
<td>1.000 007</td>
<td>1.000 004</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td>1.000 000</td>
<td></td>
</tr>
</tbody>
</table>

Table 5  Approximation process by approximation algorithm 1, where $X$ is the estimated configuration, $X_0$ is the initial, $X_1$ is the first, and $X_i$ is the $i$th calculated instantaneous configurations of the moving platform; $\phi_x$, $\phi_y$ and $\phi_z$ are a rotation of around the fixed $x$, $y$ and $z$ axes, respectively

<table>
<thead>
<tr>
<th>$X$</th>
<th>$x$/m</th>
<th>$y$/m</th>
<th>$z$/m</th>
<th>$\phi_x$/rad</th>
<th>$\phi_y$/rad</th>
<th>$\phi_z$/rad</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_0$</td>
<td>2.000 000</td>
<td>0.000 000</td>
<td>1.860 000</td>
<td>0.000 000</td>
<td>0.000 000</td>
<td>0.000 000</td>
</tr>
<tr>
<td>$X_1$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X_2$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X_3$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X_4$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X_5$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X_6$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
<tr>
<td>$X$</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
<td>Complex number</td>
</tr>
</tbody>
</table>
Table 6  Approximation process of slider position variable $q$ corresponding to Table 5 by approximation algorithm 1, where $q_0$ is the initial position, $q_1$ is the first and $q_i$ is the $i$th calculated instantaneous positions, and $q_{0j}(j=1, 2, \ldots, 8)$ is the position of the $j$th slider.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$q_{01}/m$</th>
<th>$q_{02}/m$</th>
<th>$q_{03}/m$</th>
<th>$q_{04}/m$</th>
<th>$q_{05}/m$</th>
<th>$q_{06}/m$</th>
<th>$q_{07}/m$</th>
<th>$q_{08}/m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>Complex numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_1$</td>
<td>Complex numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_i$</td>
<td>Complex numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q$</td>
<td>Complex numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusions

This paper presents the forward displacement analysis of the 8-PSS redundant parallel manipulator. To avoid the extraction of root in the approximation process, the forward displacement analysis of the 8-PSS redundant parallel manipulator is transformed into another equivalent problem on the assumption that the strut is extensible while the slider is fixed. The problem is solved by a modified approximation algorithm which predicates that the manipulator will move along a pose vector to reduce the difference between the desired configuration and an instantaneous one, and the best movement should be with minimum norm and least quadratic sum. The difference between the instantaneous configuration and the desired one is taken as the approximation step. In this way, the convergence domain of the approximation algorithm is enlarged. The simulation results demonstrate the efficiency of the proposed algorithm. The algorithm is general and can be applied to solving the forward displacement analysis of the redundant parallel manipulator actuated by a revolute joint.

Acknowledgements

The author would like to thank the anonymous reviewers for their very useful comments.

References


Edited by LUO Min