Real-Time Solutions to the Forward Kinematics of a 2RSS + RRR Parallel Mechanism

Hongdong Zhang, Yuru Zhang and Dangxiao Wang

Abstract In this paper, we propose a new parallel mechanism designed for haptic interface. The haptic device consists of a 2RSS + RRR parallel mechanism with 3 degrees of freedom. In general, the control of haptic devices requires a high servo rate up to 1 kHz which demands a fast and real-time solution to the forward kinematics. Newton–Raphson method is one of the most efficient solutions to achieve real time requirement. However, its efficiency relies deeply on initial value of iterations. We present a methodology to overcome this limitation. We first model the forward kinematics of 2RSS + RRR as an 8th-degree polynomial equation in one unknown. We then propose two methods for determining the initial value to reduce the iterations and computing time. The numerical examples in the paper demonstrate that average time of 0.15 ms (6.7 kHz) for the solutions is achieved with an accuracy of 0.001 mm. The methodology proposed in this paper is general and can be applied to other applications requiring real time solutions to forward kinematics of parallel mechanisms.

Keywords Parallel mechanism · Forward kinematics · Real-time · Newton-Raphson

1 Introduction

Parallel mechanisms have found many applications in different areas, such as aircraft simulation, force-torque sensor, CNC machine, haptic device, etc. It is well known that forward kinematics of parallel mechanisms is, in general, highly nonlinear and difficult to solve in real time. This issue is specifically challenging when a parallel mechanism is used for haptic devices. This is because the control of haptic devices often requires high update rate. In order to provide realistic force feedback, a common practice in the design of haptic devices is to set the update rate of the

H. Zhang · Y. Zhang (✉) · D. Wang
Beihang University, No. 37 Xueyuan Rd., Haidian, Beijing, China
e-mail: yuru@buaa.edu.cn

X. Zhang et al. (eds.), *Mechanism and Machine Science*,
Lecture Notes in Electrical Engineering 408,
DOI 10.1007/978-981-10-2875-5_82

control loop as 1 kHz [1, 2], which requires that the solution to the forward kinematics of a parallel mechanism much be found within less than 1 ms.

Various numerical methods have been applied to solve forward kinematics of parallel mechanisms [3–5]. A fast, robust and practical algorithm was presented specially to solve the forward kinematics of Stewart Platform [3]. The Newton-Raphson method was modified to overcome the tendency to fail when the constraint equations become poorly conditioned. Based on the simultaneous solution of three constraint equations, 0.2 ms were taken at regular configurations and 0.22 ms when the platform was near to a singularity. Another modified Newton-Raphson method [4] based on Taylor's series was proposed and the solutions could be obtained requiring just a few interaction steps. However, calculation time was not mentioned in the paper.

Many effects have been made to convert the nonlinear equations of forward kinematics to high degree polynomial with one unknown so that forward kinematics solutions could be obtained more quickly [6–9]. A mono-dimensional-search algorithm was reported to the forward kinematics solution of the general 6-6 fully parallel mechanism in [6]. All the real solutions, free from extraneous, of the forward kinematics could be found out through this method relied on a high degree polynomial in one unknown. A kinematic mapping, i.e., to map three-dimensional motions into a seven-dimensional quasi-elliptic space, was introduced in [7] and finally a univariate 40th-degree polynomial was obtained. A quaternion to represent the transformation matrix was introduced in [8]. A concise closed-form solution to the forward kinematics of the Stewart platform was obtained and as a result, only univariate quadratic equations were required to solve. Gröbner bases were used by Gan [9] to analyze the forward kinematics of the general 6-6 Stewart mechanism, which was reduced to a 40th-degree polynomial equation in one unknown. However, calculation time was not mentioned in these papers.

Other techniques to find forward kinematics solutions include neural networks, genetic algorithm and hybrid strategy. Yee [10] used a BP network to recognize the relationship between input values and output of the forward kinematics problem. It took about 1 ms to find the forward kinematics solution with an average accuracy of 0.009 units by performing several iterations. However, long hours were required for training before implementation. A genetic algorithm was presented in [11]. Although, this method converged to a solution within a broader search domain compared to the Newton-Raphson scheme, it took about 12–15 times longer, average 10 ms, than Newton-Raphson method to find the forward kinematics solution. Kang [12] employed the RBF neural network, which has a universal approximation capability, to model the forward kinematics of a hybrid structure robot. Although this method avoids the geometric parameters measurement in the real robot, a relative long time (6.84 ms) was spent on obtaining solutions to the forward kinematics with an average position error of 0.0986 mm. A hybrid strategy to solve forward kinematics in parallel manipulators was reported in [13]. A modified form of multilayered perceptron with back propagation learning was used to predict the initial position of the forward kinematic for the standard Newton–Raphson numerical technique. The hybrid strategy could achieve an accuracy of about 0.01 mm and 0.01° in the position and orientation parameters in less than two iterations and 20 ms of execution time.

In this paper, we propose a new 3-DoF parallel mechanism for a haptic device, which has two RSS chains and one RRR chain. To achieve the frequency of control loop higher than 1 kHz, we model the forward kinematics of the parallel mechanism as an 8th-degree polynomial equation in one unknown. Newton–Raphson method is applied and the initial value of iterations is carefully studied to obtain highly efficient and accurate solution of the forward kinematics.

2 Description for the 2RSS + RRR Parallel Mechanism

The schematic representation of the 2RSS + RRR parallel mechanism is illustrated in Fig. 1. This mechanism includes two RSS limbs (R: revolute joint; S: spherical joint) with an actuator at the first revolute joint for each limb and one RRR limb with an actuator at the first revolute joint. The geometric characteristics associated with the components of all limbs are as follows (the nomenclature is showed in Table 1): A plane p can be determined by C, O_0C, O_1O_2 and the axes of the joints R_1 and R_2, in which the line $(O_1O_2) \perp (O_0C)$ and the axes of R_1 and R_2 are parallel to O_0C. The axis of R_3 is parallel to O_1O_2 and has a distance of c from O_3 to the plane p. The points of O_1, O_2, M_1 and M_2 are in the same plane. The axis of R_4 is parallel to R_3 and simultaneously perpendicularly intersects the axis of R_5. It should be noted that a U (Universal) joint is assimilated by R_4 and R_5. Besides, the line $(P_1P_2) \perp (QM_3)$.

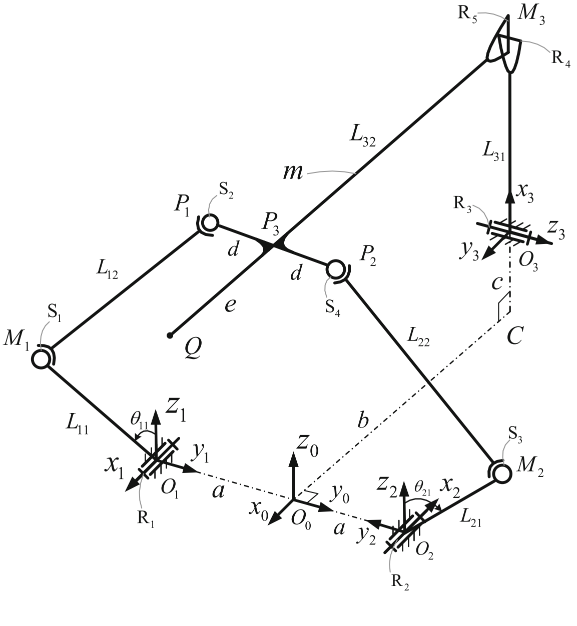


Fig. 1 The 2RSS + RRR parallel mechanism

Table 1 Nomenclature

Letters	Definitions
m	The moving platform
Q	The reference point fixed on the moving platform
R_i, $i = 1, 2, 3, 4$	The revolute joints
S_i, $i = 1, 2, 3$	The spherical joints
P_i, M_i, $i = 1, 2$	The center point of the joints S_i
M_3	The intersection of axes of R_4 and R_5
P_3	The intersection of P_1P_2 and MQ
a	The length of O_0O_1 and O_0O_2
b	The length of O_0C
c	The length of CO_3
d	The length of P_1P_3 and P_2P_3
e	The length of P_3Q
L_{i1}, $i = 1, 2, 3$	The length of O_iM_i
L_{i2}, $i = 1, 2, 3$	The length of M_iP_i
L_{33}	The length of M_3Q
O_0-$x_0y_0z_0$	The coordinate system fixed on the ground
O_i-$x_iy_iz_i$, $i = 1, 2, 3$	The coordinate systems fixed on the joints R_i ($i = 1, 2, 3$)
O_{3i}-$x_{3i}y_{3i}z_{3i}$, $i = 1, 2, 3$	The D-H coordinate systems of the RRR limb
O_Q-$x_Qy_Qz_Q$	The coordinate system of Q fixed on m
${}^{q}_{p}T$	The homogeneous coordinate transformation from p to q
θ_{i1}, $i = 1, 2, 3$	The rotational angles of R_1, R_2 and R_3
θ_{32}, θ_{33}	The rotational angles of R_4 and R_5
$s\theta$, $c\theta$	$\sin\theta$, $\cos\theta$
//, ⊥	Parallel constraint and perpendicular constraint

To determine mathematically the relative positions of all limbs, three coordinate systems O_i-$x_iy_iz_i$ ($i = 1, 2, 3$) of the limbs and a coordinate system O_0-$x_0y_0z_0$ fixed on the ground are established, as Fig. 1 shows. The homogeneous coordinate transformation matrices from O_0-$x_0y_0z_0$ to O_i-$x_iy_iz_i$ are

$$ {}^{O_1}_{O_0}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{O_2}_{O_0}T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{O_3}_{O_0}T = \begin{bmatrix} 0 & 1 & 0 & -b \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Consisting of a revolute joint and two spatial joints, the first and second limbs, which have more than six degrees of freedom, apply no constraint to the moving platform m. However, the third limb, which has three revolute joints and three degrees of freedom, applies three independent constraints to m. Consequently, through the graphical approach [14], we can conclude that the moving platform m has three degrees of freedom.

3 Forward Kinematics

In this section, we modeled the forward kinematics of the 2RSS + RRR parallel mechanism. Specially, for the haptic device, we only need to obtain the coordinate of Q when given the angles of active joints. To achieving this goal, we firstly led the first constraint equation by modeling the kinematics of the RRR limb on D-H method. Then, another two constraint equations were led based on the kinematics of the two RSS limbs. An 8th-degree polynomial equation in one unknown for the forward kinematics of the 2RSS + RRR parallel mechanism was finally obtained. Upon solving the one unknown of the high-order polynomial equation, the coordinate of Q will be easily calculated.

Firstly, we focus our attention to the kinematics model of the RRR limb. Referring to Fig. 2, the D-H parameters of the third limb, which are given in Table 2, can be obtained. Applying the D-H convention, the four transformation matrices are led as follows:

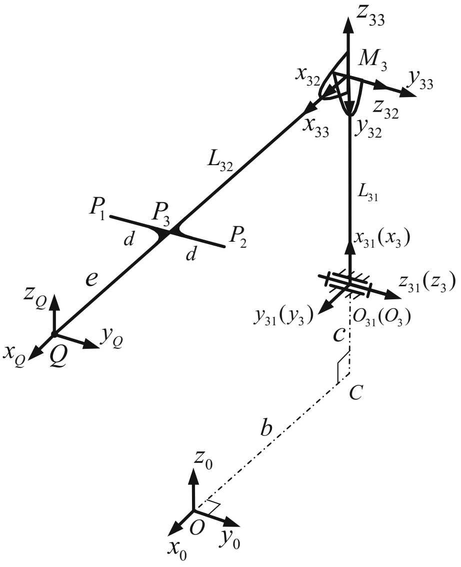


Fig. 2 D-H coordinate systems of the RRR limb

Table 2 D-H parameters of the RRR limb

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_{31}
2	L_{31}	0	0	θ_{32}
3	0	90°	0	θ_{33}
4 (Q)	L_{32}	0	0	0

$$\begin{array}{ll} {}^{O_{31}}_{O_3}T = \begin{bmatrix} c\theta_{31} & -s\theta_{31} & 0 & 0 \\ s\theta_{31} & c\theta_{31} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^{O_{33}}_{O_{32}}T = \begin{bmatrix} c\theta_{33} & -s\theta_{33} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_{33} & c\theta_{33} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^{O_{33}}_{O_{32}}T = \begin{bmatrix} c\theta_{33} & -s\theta_{33} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_{33} & c\theta_{33} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^{O_{34}}_{O_{33}}T = \begin{bmatrix} 1 & 0 & 0 & L_{32} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{2}$$

Thus, the forward kinematics equations for the third limb, which are with respect to the joints variables and the design variables, can be obtained:

$$ {}^{Q}_{O_0}T = {}^{O_3}_{O_0}T\,{}^{O_{31}}_{O_3}T\,{}^{O_{32}}_{O_{31}}T\,{}^{O_{33}}_{O_{32}}T\,{}^{Q}_{O_{33}}T \tag{3}$$

where ${}^{Q}_{O_0}T$ is the homogeneous coordinate transformation matrix from the coordinate system of O_0-$x_0y_0z_0$ to O_Q-$x_Qy_Qz_Q$, from which we can obtain the first constraint equation of the forward kinematics:

$$\begin{cases} x = L_{31}s\theta_{31} + L_{33}s(\theta_{31} + \theta_{32})s\theta_{33} - b \\ y = L_{33}s\theta_{33} \\ z = L_{31}c\theta_{31} + L_{33}c(\theta_{31} + \theta_{32})c\theta_{33} + c \end{cases} \tag{4}$$

where (x, y, z) is the coordinate of Q in O_0-$x_0y_0z_0$.

To calculate x, y and z from (4), the values of θ_{32} and θ_{33} need to be calculated first. Thus, the other two following constraint equations are led:

$$\begin{cases} L_{12}^2 = |P_1M_1|^2 \\ L_{22}^2 = |P_2M_2|^2 \end{cases} \tag{5}$$

where $|P_iM_i|$ (i = 1, 2) are the distances between P_i and M_i.

Referring to Fig. 1, the coordinates of M_i (x_{Mi}, y_{Mi}, z_{Mi}) (i = 1, 2) can be expressed as follows:

$$\begin{bmatrix} x_{M1} \\ y_{M1} \\ z_{M1} \end{bmatrix} = \begin{bmatrix} 0 \\ -a - L_{11}s\theta_{11} \\ L_{11}c\theta_{11} \end{bmatrix}, \begin{bmatrix} x_{M2} \\ y_{M2} \\ z_{M2} \end{bmatrix} = \begin{bmatrix} 0 \\ a + L_{21}s\theta_{21} \\ L_{21}c\theta_{21} \end{bmatrix} \tag{6}$$

According to the D-H parameters of the RRR limb shown in Table 1, the coordinates of P_i (x_{Pi}, y_{Pi}, z_{Pi}) (i = 1, 2) in O_0-$x_0y_0z_0$ can be obtained:

$$\begin{bmatrix} x_{P1} \\ y_{P1} \\ z_{P1} \end{bmatrix} = \begin{bmatrix} L_{31}s\theta_{31} - b + ds\theta_{33}s(\theta_{31}+\theta_{32}) + L_{32}c\theta_{33}s(\theta_{31}+\theta_{32}) \\ L_{32}s\theta_{33} - dc\theta_{33} \\ c + L_{31}c\theta_{31} + ds\theta_{33}c(\theta_{31}+\theta_{32}) + L_{32}c\theta_{33}c(\theta_{31}+\theta_{32}) \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} x_{P2} \\ y_{P2} \\ z_{P2} \end{bmatrix} = \begin{bmatrix} L_{31}s\theta_{31} - b - ds\theta_{33}s(\theta_{31}+\theta_{32}) + L_{32}c\theta_{33}s(\theta_{31}+\theta_{32}) \\ L_{32}s\theta_{33} + dc\theta_{33} \\ c + L_{31}c\theta_{31} - ds\theta_{33}c(\theta_{31}+\theta_{32}) + L_{32}c\theta_{33}c(\theta_{31}+\theta_{32}) \end{bmatrix} \tag{8}$$

Substituting all the coordinates (6), (7) and (9) into Eq. (5), we can get

$$\begin{cases} k_1 + k_2s\theta_{33} + k_3c\theta_{33} + c\theta_{32}(k_4c\theta_{33} + k_5s\theta_{33}) + s\theta_{32}(k_6c\theta_{33} + k_7s\theta_{33}) = 0 \\ r_1 + r_2s\theta_{33} + r_3c\theta_{33} + c\theta_{32}(r_4c\theta_{33} + r_5s\theta_{33}) + s\theta_{32}(r_6c\theta_{33} + r_7s\theta_{33}) = 0 \end{cases} \tag{9}$$

where k_i and r_i (i = 0–7) are with respect to θ_{11}, θ_{21}, θ_{31} and other design variables.

Equation (10) can be derived from (9) as follows:

$$\begin{cases} w_1 + w_2c\theta_{32} + w_3s\theta_{32} = 0 \\ v_1 + v_2c\theta_{32} + v_3s\theta_{32} = 0 \end{cases} \tag{10}$$

where all of w_i and v_i share a common factor of θ_{33}.

If $\theta_{11} = \theta_{21}$, we can know $\theta_{31} = 0$ by symmetry of the parallel mechanism. If $\theta_{11} \neq \theta_{21}$, we can lead the 8th-degree polynomial equation of the forward kinematics from Eq. (10) by applying the tangent half-angle substitution, $t = \tan(\theta_{33}/2)$, as follows:

$$\sum_{i=0}^{8} q_i t^i = 0 \tag{11}$$

where q_i (i = 0–8) are in terms of θ_{11}, θ_{21}, θ_{31} and other design variables.

The variable of t in Eq. (11) can be calculated through the Newton-Raphson approach which is discussed in Sect. 4. Upon obtaining t, it turns out that θ_{33}, θ_{32} can be obtained:

$$\begin{aligned} &\theta_{33} = 2\arctan(t) \\ &\theta_{32} = \begin{cases} cos^{-1}\left(\frac{w_3v_1 - w_1v_3}{w_2v_3 - w_3v_2}\right), & when \quad \theta_{11} \neq \theta_{21} \\ \arccos\left(\frac{-2\alpha\beta + \sqrt{(2\alpha\beta)^2 - 4(\alpha^2+1)(\beta^2-1)}}{2(\alpha^2+1)}\right), & when \quad \theta_{11} = \theta_{21} \end{cases} \end{aligned} \tag{12}$$

where $\alpha = -w_2/w_3$, $\beta = -w_1/w_3$.

In (12), the extraneous root of θ_{32} is avoided. Consequently, substituting θ_{11}, θ_{21}, θ_{31}, θ_{32}, θ_{33} and other design variables into Eqs. (3) and (4), we can obtain the coordinate of the reference point Q.

Concerns should be raised here that an important but also difficult problem is how to solve Eq. (11) rapidly and accurately. Although the Newton-Raphson method, one of the most efficient solutions, can be used, its efficiency relies deeply on initial value of iterations. To overcome this limitation, we proposed two methods of determining the initial value of iterations, which are described in the next section.

4 Analysis of Real-Time Forward Kinematics

Aiming to reduce the iterations and computing time, we describe two methods for determining the initial value of iterations in the Newton-Raphson method. The first method uses the solution of the previous step as the initial value to calculate t ($t = \tan(\theta_{33}/2)$). The second method determines the initial value of iterations according to the symmetry of this parallel mechanism. In the light of simulation results, the pros and cons of the two methods are discussed in the following sections. The values of all parameters for simulations are as follows: maximum velocity of Q $v_Q = 2$ m/s, servo rate $f = 1$ kHz, $a = 50$ mm, $b = 135.5$ mm, $c = 45$ mm, $d = 15$ mm, $e = 50$ mm, $L_{11} = L_{21} = 70$ mm, $L_{12} = L_{22} = 164$ mm, $L_{31} = 100$ mm, $L_{32} = 135.5$ mm, $L_{33} = 185.5$ mm, $0° \leq \theta_{11} \leq 103°$, $0° \leq \theta_{21}$ $103°$, $-20° \leq \theta_{31} \leq 20°$, $0° < \theta_{32} < 180°$, $-90° < \theta_{33} < 90°$, $-1 < t < 1$.

4.1 *First Method*

To test the validity of the first method, simulations whose results are partly shown in Fig. 3 are conducted with Matlab. We firstly use the coordinates of points on designed trajectories to calculate angles of actuators based on the inverse kinematics. Considering the maximum velocity of $v_Q = 2$ m/s and the servo rate $f = 1$ kHz, the distance between adjacent two points on the designed trajectories should be 2 mm. We then use the calculated angles of actuators to obtain solutions to the forward kinematics by the first method. The efficiency of the first method can be validated by comparing coordinates of designed trajectories with coordinates of calculated points. Most of the simulated trajectories can be tracked correctly by the first method, except the two kinds of trajectories shown in Fig. 3.

The trajectory in Fig. 3a is a line which can be described as: the y coordinate varies from −30 to 30 with an interval of 2. The x is 50 and z 145. The coordinate of starting point is (50, −30, 145). As Fig. 3a illustrates, the trajectory fails to be tracked upon passing through the plane of YOZ. The wrong solutions are far away from the trajectory. In fact, we found that all of simulated trajectories which go

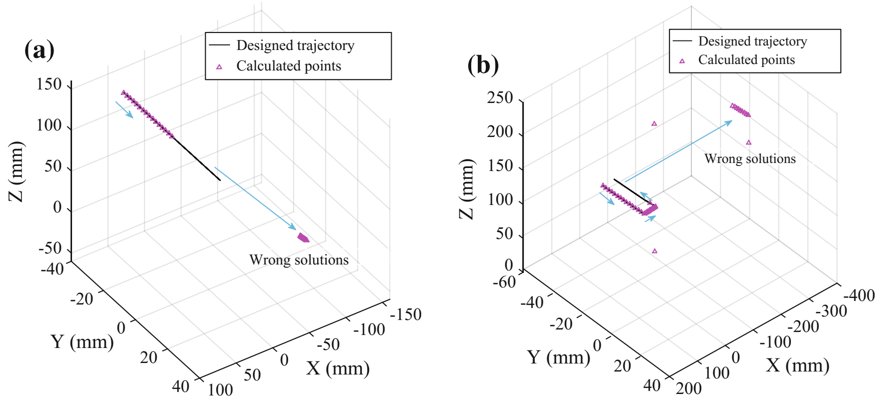


Fig. 3 Simulations of two trajectories which have wrong solutions by the first method: **a** miscalculated when going through the plane of *YOZ*; **b** wrong solutions exist when moving away from where it is close to the plane of *YOZ*

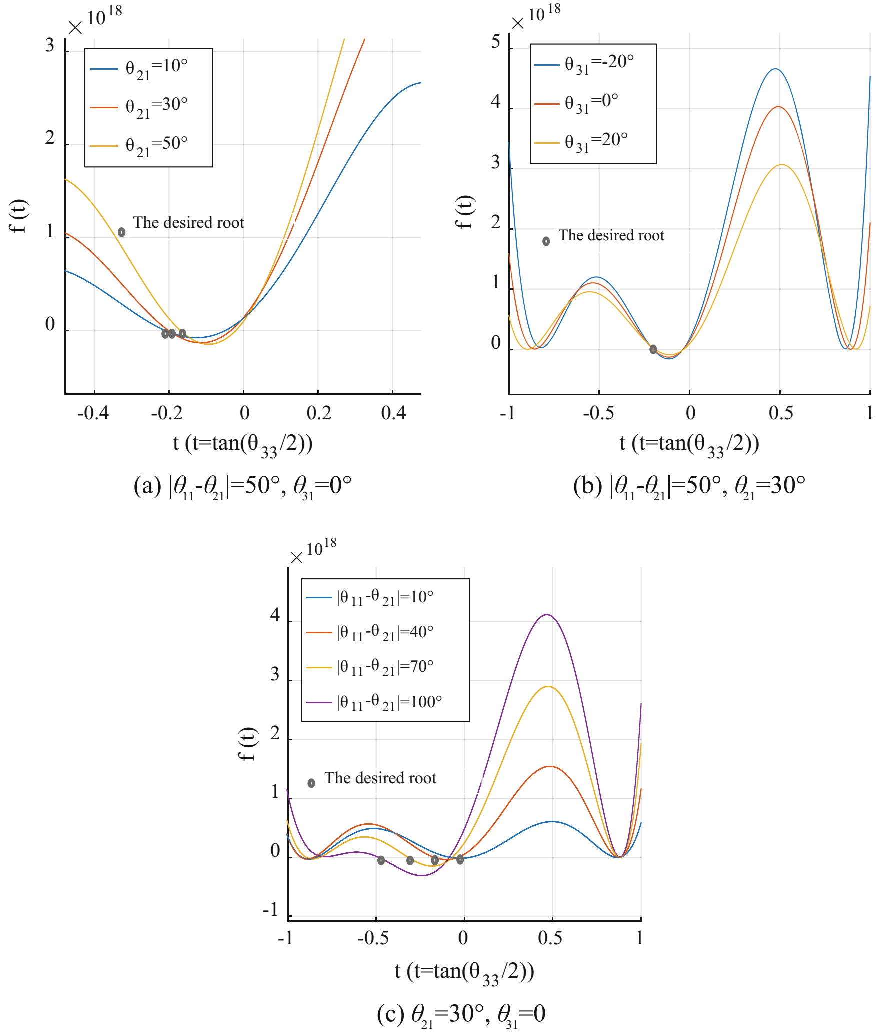
through the plane of *YOZ* are miscalculated. Once one solution is wrong, all of the following points will fail to be tracked.

Wrong solutions also occur in the second kind of trajectories which are like “U” as Fig. 3b shows. The designed trajectory, which is very close to but not through the plane at the bottom of “U”, is tracked successfully when moving to the plane of *YOZ*. However, miscalculations exist when moving along the negative *Y* axis away from where the trajectory is very close to the plane of *YOZ*.

Based on the simulation results above, we can conclude that the first method works at most of points but fails in some cases, e.g. when the trajectories pass through or move away from the plane of *YOZ*. This result is related to the symmetry of the parallel mechanism and the characters of Eq. (11). The reasons are discussed in the next section.

4.2 Second Method

The second method determines the initial value by the value of $|\theta_{11} - \theta_{21}|$, which is based on the characters of Eq. (11) partly shown in Fig. 4. Equation (11) is an equation of t, θ_{11}, θ_{21}, θ_{31} and other designed variables, in which t ($t = \tan(\theta_{33}/2)$) is the one unknown and the coefficients of the equation are determined by other variables. There are three characters of f(t) making the second method work. As one example of the first character shown in Fig. 4a, we found that when given the value $|\theta_{11} - \theta_{21}|$ and θ_{31}, the desired roots of all curves are very close to each other even though θ_{21} varies from 10° to 50° (correspondingly θ_{11} varies from 60° to 100°). This character is not limited just to the values of $|\theta_{11} - \theta_{21}|$ and θ_{31}. Figure 4b shows one case of the second character that the desired root is slightly affected



(a) $|\theta_{11}$-$\theta_{21}|$=50°, θ_{31}=0°

(b) $|\theta_{11}$-$\theta_{21}|$=50°, θ_{21}=30°

(c) θ_{21}=30°, θ_{31}=0

Fig. 4 Distribution of solutions of Eq. (11): **a**, **b** solutions depend slightly on θ_{21} and θ_{31}; **c** solutions depend largely on $|\theta_{11} - \theta_{21}|$. The smaller $|\theta_{11} - \theta_{21}|$ is, the nearer to zero the solution gets

when the values of $|\theta_{11} - \theta_{21}|$ and θ_{21} are given although θ_{31} varies over a wide range from −20° to 20°.

The third character should also be noted that there is always another undesired root close to the desired root which is generally approximate to zero, as Fig. 4c illustrates. Furthermore, the smaller the value of $|\theta_{11} - \theta_{21}|$ is, the more close to each other the two roots are. As a result, both of the roots are more approximate to zero. That is why wrong solutions often exist by the first method when the

Table 3 Initial value of t ($t = \tan(\theta_{33}/2)$)

$\lvert\theta_{11} - \theta_{21}\rvert$ (°)	t_0 ($\theta_{11} < \theta_{21}$)	t_0 ($\theta_{11} > \theta_{21}$)
(0, 1]	0.00873	−0.00873
(1, 2]	0.01746	−0.01746
(2, 5]	0.04366	−0.04366
(5, 10]	0.08748	−0.08748
(10, 20]	0.13165	−0.13165
(20, 30]	0.17633	−0.17633
(30, 40]	0.22169	−0.22169
(40, 60]	0.26795	−0.26795
(60, 90]	0.31530	−0.31530
(90, 100]	0.36397	−0.36397

trajectories are close to or go through the plane of *YOZ*. When the value of $\lvert\theta_{11} - \theta_{21}\rvert$ is very small, Q is close to the plane of *YOZ* and the two roots are approximate. If the variation of θ_{33} (correspondingly means t) is large during a servo cycle, wrong solutions are usually obtained because the initial value (the solution of the previous step) is more close to the undesired root.

Based on the three characters above, the initial values of t_0 for Newton-Raphson method according to the value of $\lvert\theta_{11} - \theta_{21}\rvert$ are listed as shown in Table 3. To validate the second method, we choose the ones which have wrong solutions (in Fig. 3) as the testing trajectories using the initial values in Table 3. As shown in Fig. 4a, both of trajectories are tracked successfully. Besides, we randomly select eighty points in the workspace of the parallel mechanism to test the second method. All of the points can be calculated correctly as Fig. 5b illustrates. It demonstrates that the second method can not only calculate continuous trajectories but also discrete points in the workspace.

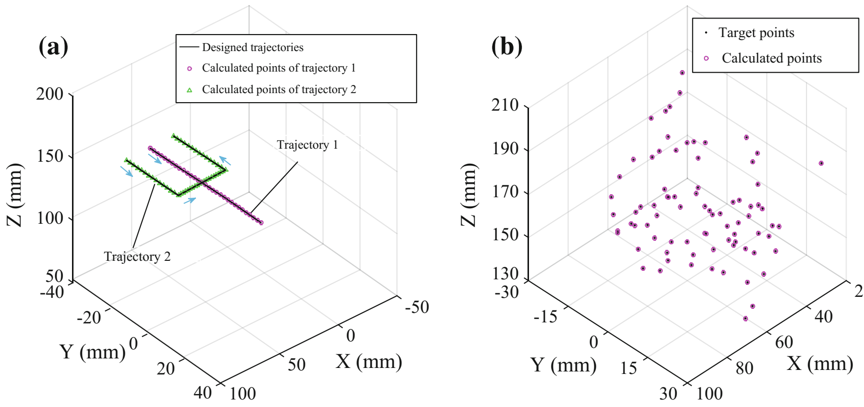


Fig. 5 Simulations of the second method: **a** two kinds of continuous trajectories both can be tracked successfully; **b** no wrong solution exists for random discrete points in the workspace

4.3 Results, Comparison and Discussion

In Sects. 4.1 and 4.2, we talked about the validity of the proposed methods of determining the initial values. In addition to this, we paid more attention to the computing time of the two methods. As known, computing time is related to configuration of CPU, initial value and tolerance of the unknown variables, etc. The simulations in this paper were all conducted on a 3-GHz computer with the Intel Core 2 Duo CPU and 4G RAM.

Considering smaller tolerance of t ($t = \tan(\theta_{33}/2)$) will cost more computing time, we conducted a simulation using the data in Fig. 5b based on the second method to find the relationship between the tolerance of t and the error of solutions. Referring to the simulation result shown in Fig. 6, we finally select 10^{-8} as the tolerance of t, which can obtain an average accuracy of 0.00099 mm and a maximum error of 0.03 mm at the these points.

Using the tolerance of 10^{-8}, we simulated the average computing time of three continuous trajectories based the two proposed methods. Comparing the results shown in Table 4, we can conclude that generally less computing time costs by the first method than the second method, especially in case that the variation of t is very

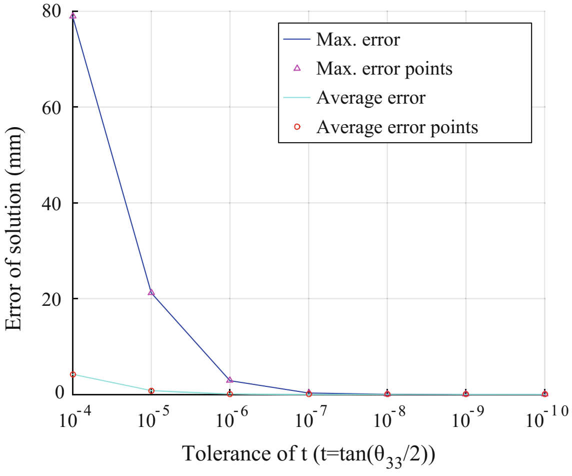


Fig. 6 Simulation of the maximum and average errors of solution with the tolerance of t: the smaller the tolerance of t is, the smaller the error of solution gets

Table 4 Comparison between the two methods

Trajectory	Method	Average time (ms)	Average iteration	Average error (mm)	Maximum error (mm)	Wrong solution
1	1st	0.12	6	1×10^{-13}	5×10^{-13}	No
	2nd	0.15	7	1×10^{-4}	2×10^{-4}	No
2	1st	0.15	7	3×10^{-5}	4×10^{-4}	Yes
	2nd	0.15	7	2×10^{-4}	1×10^{-3}	No
3	1st	0.025	2	9×10^{-14}	2×10^{-13}	No
	2nd	0.12	6	9×10^{-14}	2×10^{-13}	No

small in a servo cycle as the trajectory 3 does. With the same tolerance of t, a higher accuracy of solutions can usually be obtained by the first method with the highest accuracy of up to 0.0002 mm.

However, as mentioned above, wrong solutions often exist by the first method in some cases which must be avoided in practical occasions. As the trajectories have to be continuous, if one solution is wrong, all of the following points will fail to be calculated correctly. Although more computing time and iterations cost, the second method are robust because right positions can be calculated without the influence of the solution of the previous step, which is quite different from the first method. Besides, even though with a lower accuracy, the second method can also work in many practical cases whose demands are not high.

It is worth considering to calculate solutions to the forward kinematics using both of the methods. For example, in the case that trajectories are far from the plane of YOZ, we can use the first method and in other cases, the second method can be used. As a result, both of a high accuracy and less computing time can be obtained in the workspace of the parallel mechanism.

5 Conclusions

We proposed a new parallel mechanism for the design of haptic devices. The forward kinematics of the parallel mechanism can be modeled as an 8th-degree polynomial equation in one unknown. To meet the control requirement of 1 kHz update rate, we proposed two methods to determine the initial value in the Newton-Raphson scheme for the forward kinematics. The first method uses the solution of the previous step as the initial value for iterations. An accuracy of about 0.0004 mm was obtained with the computing time of 0.15 ms. However, wrong solutions existed in some occasions. The second method determines the initial value according to the value of $|\theta_{11} - \theta_{21}|$. This method is robust comparing to the first method. The computing time same with the first method was obtained but with a lower accuracy of 0.001 mm. Higher computing efficiency and accuracy may be obtained by combining both of the two methods.

References

1. Mohand-Ousaid A, Millet G, Regnier S, Haliyo S, Hayward V (2012) Haptic interface transparency achieved through viscous coupling. Int J Robot Res 31(3):319–329
2. Zhang YR, Li CB, Wang DX et al (2015) Task oriented method for designing haptic devices. J. Mech Eng 51(13):212–217
3. McAree PR, Daniel RW (1996) A fast, robust solution to the stewart platform forward kinematics. J Rob Syst 13(7):407–427
4. Ku DM (1999) Direct displacement analysis of a Stewart platform. Mech Mach Theory 34:453–465

5. Akcali ID, Mutlu H (2006) A novel approach in the direct kinematics of Stewart platform mechanisms with planar platforms. ASME J Mech Des 128:252–263
6. Innocenti C, Parenti-Castelli V (1993) Forward kinematics of the general 6-6 fully parallel mechanism: an exhaustive numerical approach via a mono-dimensional-search algorithm. ASME J Mech Des 115:932–937
7. Husty ML (1996) An algorithm for solving the direct kinematics of general Stewart-Gough platforms. Mech Mach Theory 31(4):365–380
8. Ji P, Wu HT (2001) A closed-form forward kinematics solution for the 6–6 Stewart platform. IEEE Trans Robot Automat 17(4):522–526
9. Gan DM, Liao QZ, Dai JS et al (2009) Forward displacement analysis of the general 6-6 Stewart mechanism using Gröbner bases. Mech Mach Theory 44:1640–1647
10. Yee CS, Lim KB (1997) Forward kinematics solution of Stewart platform using neural networks. Neurocomputing 16:333–349
11. Boudreau R, Turkkan N (1996) Solving the forward kinematics of parallel manipulators with a genetic algorithm. J Rob Syst 13(2):111–125
12. Kang RJ, Chanal H, Dai JS, Ray P (2015) Comparison of numerical and neural network methods for the kinematic modeling of a parallel-serial structure robot. J Mech Eng Sci, Proc Inst Mech Eng, Part C 229(6):1162–1171
13. Parikh PJ, Lam SSY (2005) A hybrid strategy to solve the forward kinematics problem in parallel manipulators. IEEE Trans Robot 21(1):18–25
14. Yu JJ, Pei X, Zong GH (2014) Graphical approach to creative design of mechanical devices. Science Press, Beijing, China