IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 8, 2019, accepted August 24, 2019, date of publication September 9, 2019,
date of current version September 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940064

An Efficient Numerical Method for Forward Kinematics of Parallel Robots

QIDAN ZHU AND ZHENG ZHANG
College of Automation, Harbin Engineering University, Harbin 150001, China
Corresponding author: Zheng Zhang (zhangzheng@hrbeu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 51379044.

ABSTRACT Solving the forward kinematics of parallel robots efficiently is important for real-time applications. However, it remains a difficult problem due to its high nonlinearity. This paper combines artificial neural networks and the Global Newton–Raphson with Monotonic Descent (GNRMD) algorithm to decrease the training sets of neural networks while avoiding divergence problem. Furthermore, simplified Newton iteration is introduced to reduce the duration of solution time. The proposed method is demonstrated taking a Stewart platform as an example and the nonlinear equations are established with the geometrical method. Based on the continuous characteristic of real-time applications, the result of the previous solution cycle is used as the initial value of the current solution cycle. Moreover, a threshold adjusting the effective scope of GNRMD algorithm and simplified Newton iteration is set to balance the efficiency and number of iteration. The performance of the algorithm is verified in the environment of Microsoft Visual Studio 2013 based on the continuous feedback of the Stewart platform. Besides, it is compared with GNRMD algorithm and a higher-order numerical method. The results indicate that the proposed algorithm can improve the efficiency of solving the forward kinematics problem.

INDEX TERMS Parallel robots, forward kinematics, numerical method, real-time application.

I. INTRODUCTION

Parallel robots have been extensively studied for decades and have been widely applied to various fields due to the following advantages: high rigidity, accuracy, and load-bearing capacity [1]–[5]. The kinematics problem is fundamental for parallel robots because it provides the mapping between Cartesian space and joint space. The inverse kinematics problem (IKP) can be easily solved geometrically. Compared with the IKP, the forward kinematics problem (FKP) is more difficult due to the high nonlinearity and the various structures of parallel robots [6] and the efficiency for solving the FKP is important for real-time applications.

Many researchers have made their contributions to solve the FKP of parallel robots. The existing methods can be classified into four categories: analytical approaches, numerical methods, intelligent algorithms, and adding extra sensors. Analytical approaches tend to find the closed-form solution [7]–[9]. It is only useful for specific structures and the derivation is sophisticated. Compared with analytical approaches, numerical methods can get an iterative solution without suffering the problem of multi-solution. Besides, they are more suitable in practice because the methods are simple and can be extended to various structures of parallel robots [10]–[13]. But the convergence performance is a key issue, e.g., convergence ability and convergence speed, impacting the behavior of control systems. Recently, intelligent algorithms for the FKP such as artificial neural networks (ANNs) [14] and Support Vector Machines (SVM) [15] are applied and can meet the accuracy requirement. But the training process weakens efficiency even under high-performance computer condition. Compared with other methods, adding extra sensors is the simplest method to obtain a unique solution with the least computational burden [16]. Unfortunately, the application range of this method is limited due to the economic cost together with the measurement and assembly error. In summary, the existing approaches for the FKP still have many limitations, especially in terms of balancing efficiency and accuracy. Among the methods mentioned above, the numerical method has the unique advantage in generality and simplicity so that it has received considerable attention [17].

Newton–Raphson (NR) algorithm is a widely used numerical method. It can approach the solution with high

The associate editor coordinating the review of this manuscript and approving it for publication was Chenguang Yang.

convergence speed. But it has the divergence problem if the initial value is far from the solution. Various improvements have been carried out to overcome the shortcoming of the NR method to make it more suitable for solving the FKP. To generate suitable initial value for the NR method, a hybrid strategy which combines ANNs with the NR method was proposed by Parikh and Lam [18]. The well-trained ANNs restrict the initial value to a certain domain so that the convergence is guaranteed and the number of iteration is reduced compared with traditional NR method. However, large sample sets are required for training of ANNs and the training process is time-consuming. The Global Newton–Raphson with Monotonic Descent (GNRMD) algorithm which is independent on the initial value was proposed by Yang *et al.* [19]. It can achieve global convergence of the NR method, but the efficiency is weakened due to the lack of limitations of the initial value. NR method is also combined with homotopy methods [20]–[22] to solve the FKP, but the long calculation time makes them not suitable for real-time applications. Therefore, the balance of accuracy and efficiency is still a challenge for solving the FKP.

In this paper, we propose a method that combines ANNs and GNRMD algorithm to solve the FKP of parallel robots efficiently. The two methods can remedy each other's shortcomings. ANNs can be trained with fewer sample sets because GNRMD algorithm can avoid the divergence problem even when the initial value is far from the solution. On the other hand, the initial value is limited within a certain domain by ANNs so that the efficiency of calculation can be improved. In the iteration procedure, frequent calculation of the Jacobian matrix and its inverse is time-consuming but it has attracted little attention [10]. On this account, we introduce simplified Newton iteration [23] with the fixed inverse of Jacobian to reduce the amount of calculation in each iteration. Then a deviation threshold is set to adjust the effective scope of GNRMD algorithm and simplified Newton iteration, thus balancing the number of iteration and efficiency. In real-time applications, the deviation of pose between adjacent cycles is small due to the high sample rate. Therefore, the state in the previous cycle can be used directly as the initial state of the current cycle, which further simplifies the procedure of solving the FKP. The proposed method in this paper is named as deviation-driven algorithm and the performance is verified with experiments.

This paper is organized as follows: Section II defines the coordinate systems and establishes the nonlinear equations of the FKP for the following analysis. The deviation-driven algorithm is described in Section III. Then experiments verifying the accuracy and efficiency of the proposed algorithm are shown in Section IV. Finally, the discussion of results and future work are given in Section V.

II. ESTABLISHMENT OF NONLINEAR EQUATIONS

The main components of the parallel robot are two platforms connected by several links. The base platform is fixed to the ground and the mobile platform usually works as the end-effector. Motions of the mobile platform can be achieved by changing the displacements of links. In this paper, we take the 6-6 Stewart platform as an example to demonstrate the proposed algorithm.

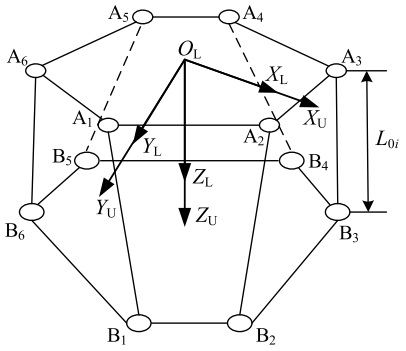

FIGURE 1. 6-6 Stewart platform and coordinate systems at initial state.

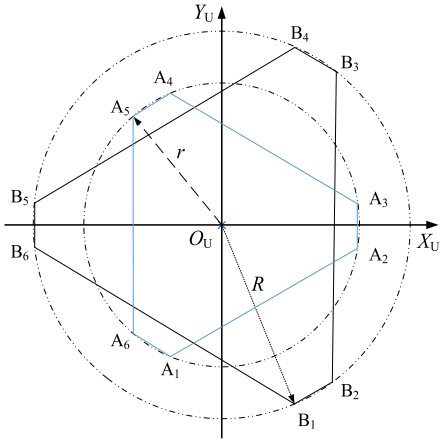

FIGURE 2. Perspective planar view at initial state.

A general structure of the 6-6 Stewart platform and its coordinate systems at initial state are illustrated in Fig. 1. The moving coordinate system $\{\mathbf{U}\}$ is fixed to the mobile platform and its origin is at the geometric center of the platform. The base coordinate system $\{\mathbf{L}\}$ remains stationary in Cartesian space at the position of $\{\mathbf{U}\}$ when the displacements of the actuators are zero (initial state). The parameter L_{0i} $(i = 1, \ldots, 6)$ is the length of the actuator between the connection point A_i and B_i at initial state. The connection points of the Hooke hinges on the mobile and base platform are distributed on circles of radius r and R respectively. A perspective planar view of the position of connection points is given in Fig. 2. The motion of the mobile platform consists of angular motions and linear motions, i.e., roll (q_1), pitch (q_2), and yaw (q_3) together with surge (q_4), sway (q_5), and heave (q_6). Therefore, the pose of $\{\mathbf{U}\}$ with respect to $\{\mathbf{L}\}$ can be defined as

$$\boldsymbol{P} = [\, q_1, \quad q_2, \quad \ldots, \quad q_6 \,]^{\mathrm{T}} \tag{1}$$

The transformation matrix of the mobile platform can be represented by ${}^{\mathrm{U}}_{\mathrm{L}}\boldsymbol{R} \in R^{3\times 3}$

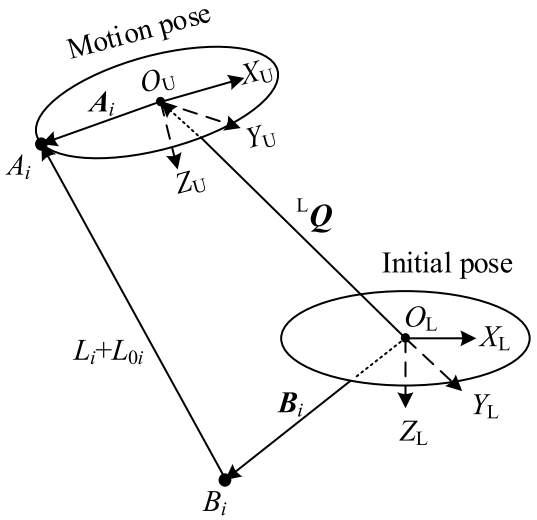

FIGURE 3. Geometric transformation of *i*th actuator.

$$
{}^{U}_{L}\boldsymbol{R} = \begin{bmatrix} cq_2cq_3 & -cq_1sq_3 + sq_1sq_2cq_3 & sq_1sq_3+cq_1sq_2cq_3 \\ cq_2sq_3 & cq_1cq_3 + sq_1sq_2sq_3 & -sq_1cq_3+cq_1sq_2sq_3 \\ -sq_2 & sq_1cq_2 & cq_1cq_2 \end{bmatrix} \tag{2}
$$

where s( ) and c( ) represent sin( ) and cos( ) respectively. The geometric transformation of the *i*th actuator in motion is shown in Fig. 3.

As shown in Fig. 3, the displacement of *i*th actuator can be solved with the geometric method and denoted as (3), which is also the IKP formula of parallel robots.

$$
L_i = f_i(\boldsymbol{P}) = \left\| {}^{U}_{L}\boldsymbol{R}\boldsymbol{A}_i + {}^{L}\boldsymbol{Q} - \boldsymbol{B}_i \right\| - L_{0i} \quad (i = 1, 2, \ldots, 6) \tag{3}
$$

f_i ( ) is the function mapping the pose from Cartesian space to the displacement of *i*th actuator L_i in joint space. ${}^{L}\boldsymbol{Q} = [q_4, q_5, q_6]^{T}$ denotes linear translation from the coordinate system {**L**} to {**U**}. $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are constant vectors of the upper and lower connection points in the coordinate system {**U**} and {**L**} respectively. L_{0i} is the constant length between the joint position of A_i and B_i when the displacements of actuators are zero. Then we can obtain the nonlinear equation set of the FKP.

$$
F_i(\boldsymbol{P}) = f_i(\boldsymbol{P}) - L_{iM} = 0 \tag{4}
$$

L_{iM} is the measured displacement of *i*th actuator. The set of (4) forms the nonlinear equations for the FKP of parallel robots. In matrix form, they can be written as

$$
\boldsymbol{F}(\boldsymbol{P}) = \boldsymbol{0} \tag{5}
$$

where $\boldsymbol{F} = \begin{bmatrix} F_1 & F_2 & \ldots & F_6 \end{bmatrix}^{T} \in R^{6\times 1}$.

III. DEVIATION-DRIVEN ALGORITHM

The FKP of parallel robots aims to solve $\boldsymbol{P}$ in (5) with a set of measured displacements of actuators. The deviation-driven algorithm consists of three parts: ANNs, GNRMD algorithm, and simplified Newton iteration. The iteration process is divided into two effective scopes by a threshold: convergence scope for GNRMD algorithm and first-order convergence scope for simplified Newton iteration. First, the ANNs generate an approximate solution corresponding to the measured displacements of actuators. Then the initial value is converged to the threshold with GNRMD algorithm. After that, the inverse of Jacobian matrix is fixed and the simplified Newton iteration proceeds until the required accuracy is satisfied. Scheme of the deviation-driven algorithm is shown in Fig. 4.

A. ANNS AND GNRMD ALGORITHM

Considering the convergence is guaranteed by the GNRMD algorithm, the accuracy of ANNs' output can be decreased to shorten the training time and to simplify the sample sets. Meanwhile, the initial value for GNRMD algorithm is restricted so that the convergence efficiency is improved. Therefore, we can obtain the balance between accuracy and efficiency.

The research of [24] has made a comparison of ANNs with different structures. Though the structure of ANNs applied in [24] achieves the separate output of position and orientation, it increases the workload during the training phase and is time-consuming because of the extra network. As a result, the adopted structure of the network in this paper is an all class one network (ACON) with six inputs and six outputs. The structure of ANNs with one hidden layer is shown in Fig. 5.

The activation functions are generally selected as the sigmoid function:

$$
g(x) = \frac{1}{1 + e^{-x}} \tag{6}
$$

In this paper, the network/data manager toolbox in MATLAB is adopted to train the ANNs. Before the training process, a certain number of poses within workspace should be selected and the corresponding displacements of actuators are obtained by solving IKP with (3). The two groups of data together form the sample sets for training ANNs, then

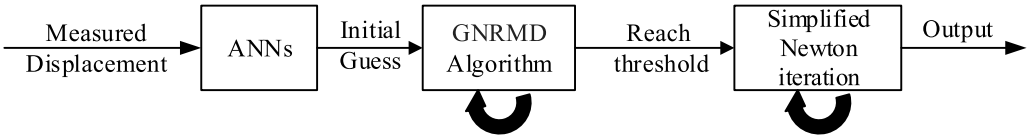

FIGURE 4. The deviation-driven algorithm.

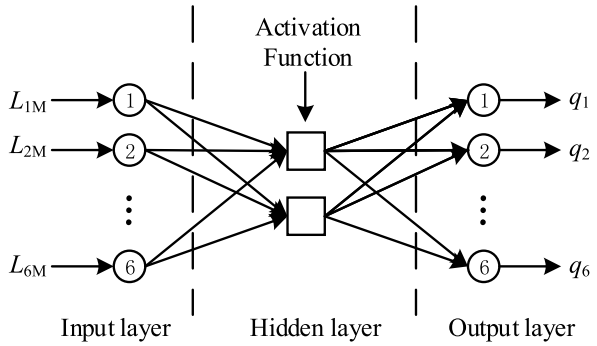

FIGURE 5. Structure of ANNs with one hidden layer.

mappings of the two groups of data are inverted. In other words, the calculated data in joint space are used as the inputs of ANNs and the corresponding pose data in Cartesian space are the expected outputs of ANNs. The network is trained with the well-known backpropagation (BP) algorithm and the details are referred to in [18].

Assume $\boldsymbol{P}_0$ generated by ANNs is the approximate solution of (5), then (5) can be expanded to (7) based on Taylor polynomial.

$$\boldsymbol{F}(\boldsymbol{P}) \approx \boldsymbol{F}(\boldsymbol{P}_0) + \boldsymbol{J}(\boldsymbol{P}_0)(\boldsymbol{P} - \boldsymbol{P}_0) \tag{7}$$

$\boldsymbol{J}$ is the Jacobian matrix, i.e., the derivative of matrix $\boldsymbol{F}$.

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial F_1}{\partial q_1} & \cdots & \frac{\partial F_1}{\partial q_6} \\ \cdots & \cdots & \cdots \\ \frac{\partial F_6}{\partial q_1} & \cdots & \frac{\partial F_6}{\partial q_6} \end{bmatrix} \tag{8}$$

Substituting (7) into (5), we can obtain the iterative formula.

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{J}^{-1}(\boldsymbol{P}_k)\boldsymbol{F}(\boldsymbol{P}_k) \quad k = 0, 1, 2, \ldots \tag{9}$$

There exist multiple structures of parallel robots. For some structures, such as redundant actuation, $\boldsymbol{J}$ is not a square matrix so the inverse Jacobian matrix does not exist. To make the proposed algorithm suitable for more structures, $\boldsymbol{J}^{-1}$ is replaced with pseudoinverse of Jacobian.

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{J}^{+}(\boldsymbol{P}_k)\boldsymbol{F}(\boldsymbol{P}_k) \tag{10}$$

Equation (10) is the formula of the NR-type method. Since its convergence is highly dependent on the accuracy of the initial value, a monotonic descent operator λ is introduced to ensure the convergence of iteration. Define $\bar{\boldsymbol{P}}_{k+1}$ as the result of (10), then a weighted average can be obtained:

$$\bar{\boldsymbol{P}}_{k+1} = \boldsymbol{P}_k - \boldsymbol{J}^{+}(\boldsymbol{P}_k)\boldsymbol{F}(\boldsymbol{P}_k) \tag{11}$$

$$\boldsymbol{P}_{k+1} = \lambda\bar{\boldsymbol{P}}_{k+1} + (1-\lambda)\boldsymbol{P}_k \quad 0 < \lambda \leq 1 \tag{12}$$

Combining (11) and (12), we have

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \lambda\boldsymbol{J}^{+}(\boldsymbol{P}_k)\boldsymbol{F}(\boldsymbol{P}_k) \quad 0 < \lambda \leq 1 \tag{13}$$

In each iteration, the selection of λ must satisfy the monotone convergence:

$$\left\|\boldsymbol{F}(\boldsymbol{P}_k - \lambda\boldsymbol{J}^{+}(\boldsymbol{P}_k)\boldsymbol{F}(\boldsymbol{P}_k))\right\| < \|\boldsymbol{F}(\boldsymbol{P}_k)\| \quad 0 < \lambda \leq 1 \tag{14}$$

An efficient method to select the appropriate value of λ is binary search. λ is gradually halved from 1 until (14) is satisfied.

It should be noted that the singularity of Jacobian matrix can lead to failure of solving the FKP of parallel robots and lose control of the hardware system. Therefore, the singularity problem is avoided in the mechanical design and trajectory planning procedure. In this paper, it is assumed that the pseudoinverse of Jacobian always exists in the workspace. The terminal condition of the GNRMD algorithm is:

$$\max |F_i(\boldsymbol{P}_k)| < \varepsilon_{\max} \tag{15}$$

where $\varepsilon_{\max}$ is the threshold entering the effective scope of simplified Newton iteration.

The GNRMD algorithm is equivalent to the NR method at $\lambda = 1$ and has high convergence rate [25]. In comparison with GNRMD algorithm, simplified Newton iteration has first-order convergence rate. It results in more iteration steps but requires less computational cost in each iteration. By changing the value of $\varepsilon_{\max}$, the effective scope for the two kinds of iteration can be adjusted, thus keeping a balance between the number of iteration and calculation efficiency.

B. SIMPLIFIED NEWTON ITERATION

Although the GNRMD algorithm has higher convergence rate in comparison with simplified Newton iteration and can approach the solution with fewer number of iteration, the advantage in terms of efficiency is no longer obvious when dealing with high-dimension nonlinear equations. In fact, frequent calculation of the Jacobian along with its pseudoinverse is time-consuming and slows down the solution procedure. To remedy this shortcoming, simplified Newton iteration is introduced to proceed the following iteration after (15) is satisfied:

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{C}\boldsymbol{F}(\boldsymbol{P}_k) \tag{16}$$

$$\boldsymbol{C} = \boldsymbol{J}^{+}(\boldsymbol{P}_\varepsilon) \tag{17}$$

$\boldsymbol{P}_\varepsilon$ is the first iterative result satisfying (15). $\boldsymbol{C}$ is the fixed pseudoinverse of Jacobian when $\boldsymbol{P}_k$ is at $\boldsymbol{P}_\varepsilon$.

A monitor checking the convergence after each iteration is necessary to avoid the divergence problem caused by the inappropriate selection of pseudoinverse of Jacobian:

$$\|\boldsymbol{F}(\boldsymbol{P}_{k+1}))\| < \|\boldsymbol{F}(\boldsymbol{P}_k)\| \tag{18}$$

The value of $\boldsymbol{C}$ remains fixed when the convergence monitor is satisfied. Otherwise, renew $\boldsymbol{C}$ with $\boldsymbol{J}^{+}(\boldsymbol{P}_k)$ and substitute it to (16). The terminal condition of simplified Newton iteration is:

$$\max |F_i(\boldsymbol{P}_k)| < \varepsilon_{\min} \tag{19}$$

$$k \leq N \tag{20}$$

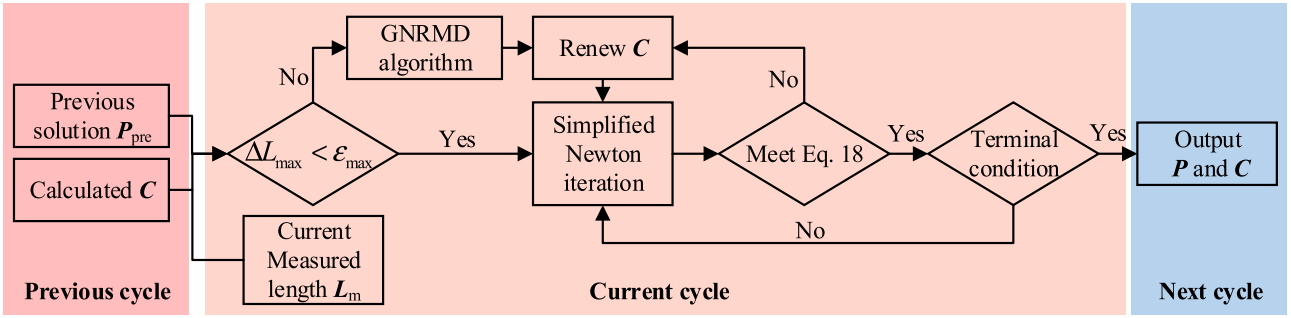

FIGURE 6. The scheme of parameter transfer among control cycles.

where $\varepsilon_{\min}$ is the permissible error (required accuracy level) of the final solution and N is the total number of iteration in one control cycle.

It should be noted that solving the FKP in real-time is more practical for parallel robots in a dynamic situation. The control cycle and sampling interval are usually at the millisecond level to continuously send commands and receive feedback. Therefore, the solutions of the FKP between adjacent control cycles are close to each other.

Define $\boldsymbol{P}_{\text{pre}}$ as the FKP solution of the previous control cycle. $\boldsymbol{P}_{\text{pre}}$ can be directly used as the initial value for the current cycle, thus saving the time of calling ANNs. If (21) and (22) are satisfied, $\boldsymbol{P}_{\text{pre}}$ can be treated as the initial value within the effective scope of simplified Newton iteration in the current cycle.

$$\Delta L_{\max} = \max \left| f_i(\boldsymbol{P}_{\text{pre}}) - L_{i\text{M}} \right| \tag{21}$$

$$\Delta L_{\max} < \varepsilon_{\max} \tag{22}$$

$\Delta L_{\max}$ is the maximum deviation of actuators' displacements between adjacent control cycles. $L_{i\text{M}}$ is the measured displacements of actuators in the current cycle. A small change in joint space causes a small change of solution, so the solution for the current cycle is within a neighborhood of the previous one. In practical applications, the steep changes of velocity in both joint space and Cartesian space between adjacent control cycles are not allowed in case of hardware damage. Therefore, the simplified Newton iteration in the current cycle can proceed using the same pseudoinverse of Jacobian transferred from the previous cycle, thus saving calculation resources. Once the deviation is larger than the threshold, GNRMD algorithm can bring the result back to the effective scope of simplified Newton iteration and repeat the above procedure. The convergence monitor also ensures the proper settings of the fixed pseudoinverse of Jacobian.

From the overall perspective of the real-time solution process, the iteration in each control cycle is driven by the deviation in joint space. The scheme of transferring pose $\boldsymbol{P}_{\text{pre}}$ and fixed pseudoinverse of Jacobian $\boldsymbol{C}$ among calculation cycles is shown in Fig. 6. $\boldsymbol{L}_{\text{m}}$ is the vector of measured displacements of actuators in the current cycle.

IV. EXPERIMENT AND RESULTS

To verify the performance of the deviation-driven algorithm, experiments were carried out based on a 6-DOF Stewart platform as shown in Fig. 7. The geometrical parameters of the Stewart platform are given in Table 1. The performance of the deviation-driven algorithm was tested in terms of accuracy, the influence of threshold, and time consumption. The terminal iteration step N was set to 10 and the permissible error was 10^{-6} cm. The following experiments were carried out in the environment of Microsoft Visual Studio 2013 and the operating system was Windows 7.

FIGURE 7. The experimental Stewart platform.

TABLE 1. Parameters of the experimental platform.

Parameters	Value
Base platform radius R (cm)	79.4
Mobile platform radius r (cm)	58.0
Length L_o (cm)	71.9
Min/Max limit of q_1 q_2 q_3 (°)	± 20.1
Min/Max limit of q_4 q_5 (cm)	± 23.6
Min/Max limit of q_6 (cm)	-39.0/0.0

TABLE 2. Mean square error with different number of neurons.

Number of neurons	Mean square error
5	4.46
10	2.85
15	0.85
20	0.49
25	0.20
30	0.19

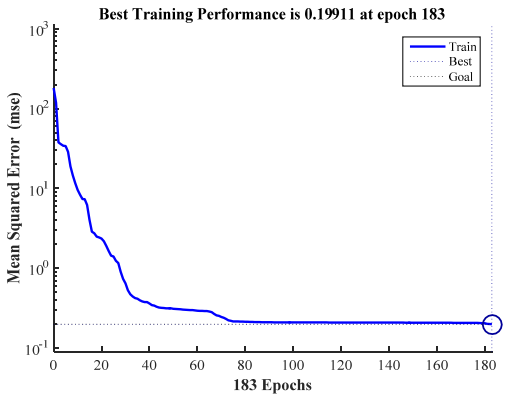

FIGURE 8. Training performance.

A. ACCURACY AND EFFICIENCY IMPROVEMENT

It is known that the initial value can determine the efficiency of the NR-type method. However, the required sample sets and training time increase with the improvement of ANNs' performance in terms of accuracy. To balance training time and accuracy of outputs, we selected ANNs with one hidden layer and the number of sample sets was 100. The hidden layer was activated by sigmoid functions and the output layer was activated by linear activation functions. Table 2 shows the mean square error of ANNs with different number of neurons in the hidden layer. The training process cost around 30 seconds. It can be seen that when the number of neurons reaches 25, there is little space for further decrease of mean square error. Therefore, we adopted the ANNs with 6-25-6 structure. The training performance is given in Fig. 8.

To verify the accuracy of the deviation-driven algorithm, 400 random sets of pose ($\boldsymbol{P}_{\text{test}}$) within the workspace of the Stewart platform were generated. They were mapped from Cartesian space to joint space via the inverse kinematics, acting as the inputs of the deviation-driven algorithm. The threshold was set to 10^{-6} cm, the same as the permissible error. Fig. 9 shows the test diagram. The error between outputs of the deviation-driven algorithm and the expected pose $\boldsymbol{P}_{\text{test}}$ are shown in Fig. 10.

It can be seen that the maximal errors are 1.58×10^{-6}° in roll and 7.86×10^{-7} cm in heave. The result confirms that the solution of the deviation-driven algorithm is of high accuracy and the divergence problem is avoided even when

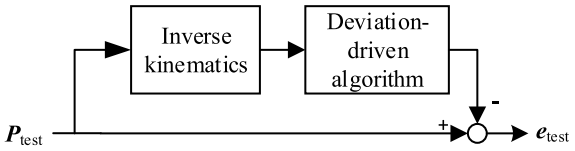

FIGURE 9. Diagram of accuracy verification.

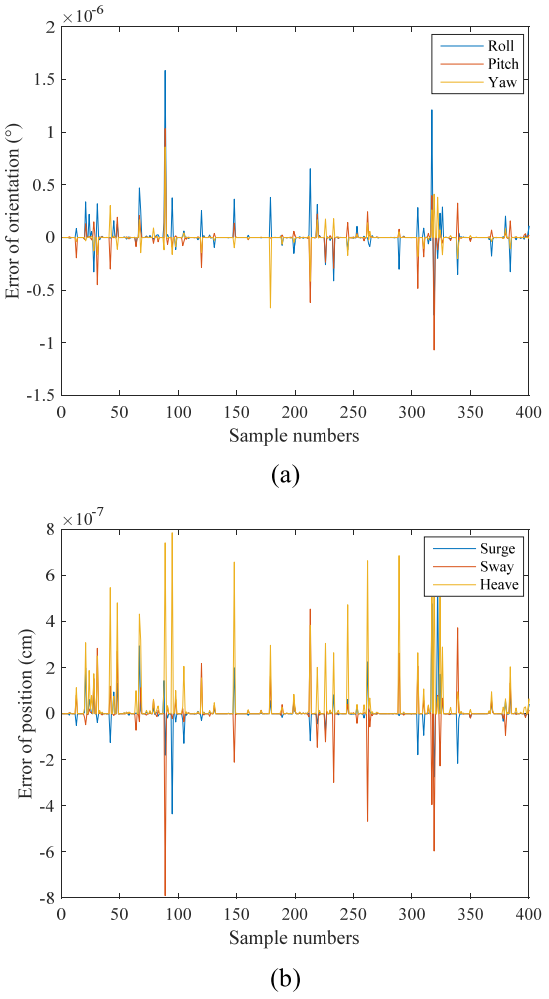

FIGURE 10. Errors of outputs. (a) Orientation error; (b) Position error.

the ANNs are trained with fewer sample sets. To demonstrate the improvement of the deviation-driven algorithm, the data in joint space corresponding to the above 400 sets of pose were used as inputs for only GNRMD algorithm and the comparison of efficiency is shown in Table 3. The initial value of GNRMD algorithm was set to $[0\ 0\ 0\ 0\ 0\ 0]^{\text{T}}$. It can be seen that the number of iteration and calculation time are both less than GNRMD algorithm when the initial value is restricted by ANNs.

Likewise, the performance of the proposed algorithm in the dynamic situation was also tested. A compound movement of the mobile platform consisting of roll (q_1) and sway (q_5) was simulated as the desired trajectory. Similar to the diagram in

TABLE 3. Comparison of efficiency.

	Average number of iteration	Required time (ms)
GNRMD algorithm	3.652	3.387
Deviation-driven algorithm	2.028	2.715

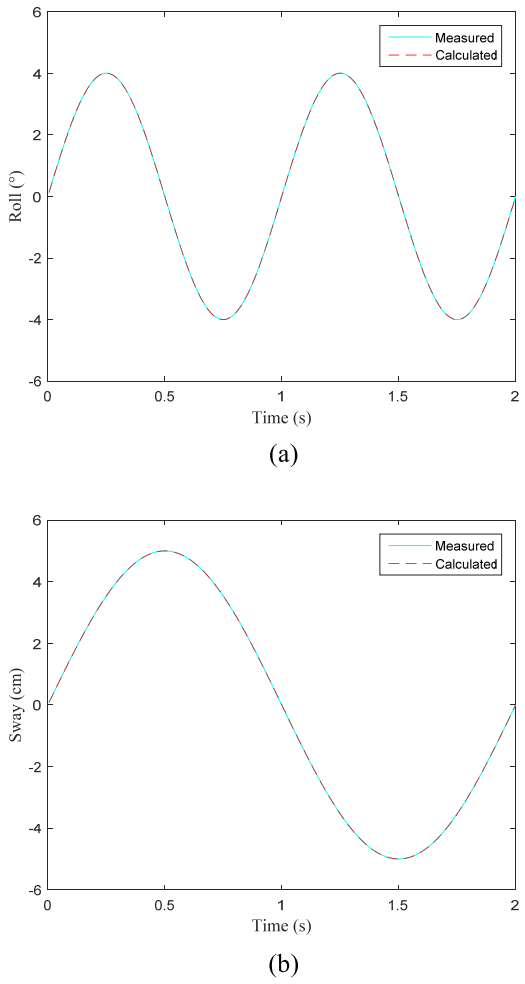

FIGURE 11. Comparison of the desired pose and calculated pose: (a) roll; (b) sway.

Fig. 9, the points on the trajectory were converted to displacements of linear actuators and transferred to the deviation-driven algorithm. The solution of jth point is directly used as the initial value for $j+1$th point.

$$\begin{cases} q_1 = 4\sin(2\pi t) & (\text{unit} : ^\circ) \\ q_5 = 5\sin(\pi t) & (\text{unit} : \text{cm}) \end{cases} \tag{23}$$

The comparison of the desired trajectory and the calculated results are shown in Fig. 11. It can be seen that the result of the deviation-driven algorithm is consistent with the expected trajectory. Fig. 10 and Fig. 11 together demonstrate that the deviation-driven algorithm is an effective method to solve the FKP.

TABLE 4. Performance with different thresholds.

Threshold	Total number of iteration / Number of simplified Newton iteration	Required time (ms)
0.001	1.830/0.845	1.352
0.005	1.895/0.979	1.297
0.010	1.997/1.169	1.273
0.050	3.643/3.642	1.557
0.100	3.643/3.642	1.557

B. INFLUENCE OF THE THRESHOLD

In this subsection, the experiments are based on real-time feedback from the Stewart platform. The displacements of linear actuators were measured by the embedded sensors in each actuator and were sent to the industrial computer in real-time. Actuators numbered 1 and 3 were driven by a sinusoidal motion command (6 cm/0.5 Hz) in joint space to generate dynamic displacements. Displacements of all actuators were set to 17.02 cm at the beginning of the motion to avoid joint limits. The feedback interval of the displacements of actuators was 2 ms. The FKP corresponding to the continuous feedback was solved on a PC with a 2.1 GHz processor and 8 GB RAM.

The threshold determines the effective scope of GNRMD algorithm and simplified Newton iteration. Due to different convergence rates of the two kinds of iteration methods, the calculation efficiency can be adjusted by the threshold. Therefore, experiments were carried out with different thresholds to test their influence on the efficiency of solving the FKP. Table 4 shows the average number of iteration and the required time to obtain the solution.

As Table 4 shows, the total number of iteration and threshold present a synchronous growth trend. When the threshold is higher than 0.05, the total number of iteration tends to be constant. Besides, the proportion of simplified Newton iteration in the total number of iteration also increases with the threshold. This trend proves that simplified Newton iteration can result in more iterations due to its lower rate of convergence compared with GNRMD algorithm. The required time shows downward trend firstly, and then upward with the increase of threshold. This phenomenon indicates that simplified Newton iteration can improve calculation efficiency while increasing the number of iteration if the threshold is small. Otherwise, the performance of the deviation-driven algorithm will be weakened due to a large number of iteration.

To further verify the improvement of the deviation-driven algorithm in terms of efficiency, NR algorithm, GNRMD algorithm and a 3rd-order numerical algorithm [24] were used as comparisons under the same environment and data. The solution of the previous cycle is used as the initial guess for the current cycle for NR algorithm. The initial value of the GNRMD algorithm was set to $[0\ 0\ 0\ 0\ 0\ -23.1]^{\mathrm{T}}$, which is close to the pose corresponding to the starting position of the mobile platform. Meanwhile, the neural networks in

TABLE 5. Performance of the comparative methods.

	Average number of iteration	Required time (ms)
NR algorithm	1.930	1.847
GNRMD algorithm	2.565	2.503
Method in [24]	1.452	2.197

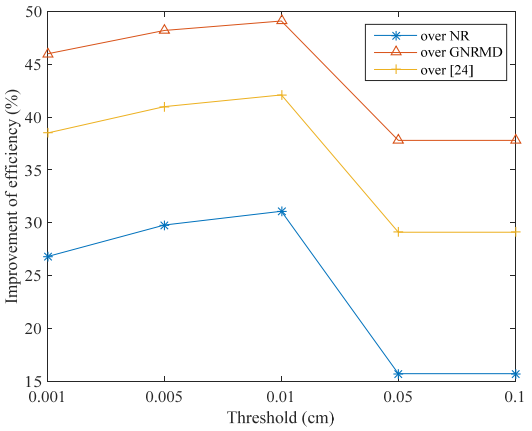

FIGURE 12. Improvement of efficiency.

[24] were trained with the same sample sets as in this paper. The permissible error of the above methods was also set to 10^{-6} cm to solve the FKP in the same precision level as the deviation-driven algorithm. Table 5 shows the average number of iteration and the required time of the two comparative methods.

Table 4 and Table 5 together confirm the efficiency of deviation-driven algorithm for real-time calculation. Fig. 12 shows the improvement of efficiency over the other three methods with different threshold. It can be seen that when the threshold is around 0.01, the deviation-driven algorithm shows the best performance.

V. DISCUSSION

The main contribution of the paper is the new deviation-driven algorithm for FKP which combines GNRMD algorithm with ANNs. Compared with only GNRMD method, the ANNs can provide the initial value for iteration so that reduce the calculation time. Compared with the hybrid strategy [18], sample sets for training can be simplified sufficiently by global convergence ability guaranteed by GNRMD algorithm. In the method of this paper, two important innovations are proposed:

1). Simplified Newton iteration is introduced to balance the efficiency and the number of iteration by the fixed inverse of Jacobian at an intermediate state of the convergence procedure so that the time consumption for updating the inverse of the Jacobian is reduced. It's important for real-time applications.

2). In the proposed algorithm, the forward kinematics and inverse of Jacobian of the previous configuration can be used directly as that of the current configuration in real-time applications if the deviation between previous and current configuration in joint space is less than a threshold, so that the ANNs calculation for obtaining initial values is needed only in the first cycle and can be simplified in later cycles.

In this paper, we solve the FKP of parallel robots with Newton-type numerical iteration. The errors of results with random inputs indicate that the proposed method can obtain the accurate solution of the FKP without divergence problem even with reduced training sets for ANNs. This is due to the global convergence guaranteed by the GNRMD algorithm. The comparison with only GNRMD algorithm proves that the efficiency is improved by restricting the initial value with ANNs. Besides, the results with both random inputs and dynamic inputs show good accuracy performances of the deviation-driven algorithm. Experiments confirm that threshold can influence the calculation time by adjusting the effective scope of GNRMD algorithm and simplified Newton iteration. Therefore, a reasonable threshold can achieve the best performance of the deviation-driven algorithm. Though the simplified Newton iteration saves computing resources in each iteration step, it will weaken the efficiency of the proposed method if the number of iteration is excessive due to a large threshold. Despite this, the required time is still shorter than that in [24] thanks to the initial value and fixed inverse of Jacobian are directly transferred from the previous cycle.

The deviation-driven algorithm is simple and easy to implement, making it convenient to extend to parallel robots of other structures. The comparative results indicate that the proposed method is more efficient even without the best performance. This method can free more computer resources for other tasks with high real-time requirements or high computational loads such as visual tracking and cooperative motion with other robots. Besides, it allows for more timely feedback in closed-loop control, thus achieving better performance in position control. The principle of the deviation-driven algorithm also provides potential methods for solving IKP of serial manipulators without closed-form solutions. The proposed method will achieve faster calculation speed in the real-time operating system as in [19]. In future work, singularity analysis will be added to the proposed method to make it suitable for parallel robots of more structures. In this paper, the threshold is manually chosen. A method adjusting threshold automatically so that the proposed algorithm can work with the best performance is valuable and will be investigated in future studies.

REFERENCES

[1] A. C. Majarena, J. Santolaria, D. Samper, and J. J. Aguilar, "Computational model for the control, performance evaluation, and calibration of a parallel mechanism," *Int. J. Adv. Manuf. Technol.*, vol. 69, nos. 9–12, pp. 1971–1979, Dec. 2013.

[2] A. Rezaei and A. Akbarzadeh, "Position and stiffness analysis of a new asymmetric 2PRR–PPR parallel CNC machine," *Adv. Robot.*, vol. 27, no. 2, pp. 133–145, Jan. 2013.

[3] J. Liu, Y. Li, Y. Zhang, Q. Gao, and B. Zuo, "Dynamics and control of a parallel mechanism for active vibration isolation in space station," *Nonlinear Dyn.*, vol. 76, no. 3, pp. 1737–1751, May 2014.
[4] V. Bevilacqua, M. Dotoli, M. M. Foglia, F. Acciani, G. Tattoli, and M. Valori, "Artificial neural networks for feedback control of a human elbow hydraulic prosthesis," *Neurocomputing*, vol. 137, pp. 3–11, Aug. 2014.
[5] S. Kumar, B. Bongardt, M. Simnofske, and F. Kirchner, "Design and kinematic analysis of the novel almost spherical parallel mechanism active ankle," *J. Intell. Robotic Syst.*, vol. 94, no. 2, pp. 303–325, May 2019.
[6] Q. Wang, J. Su, Z. Lv, L. Zhang, H. Lin, and G. Xu, "Efficient hybrid method for forward kinematics analysis of parallel robots based on signal decomposition and reconstruction," *Adv. Mech. Eng.*, vol. 9, no. 5, pp. 1–14, May 2017.
[7] T.-Y. Lee and J.-K. Shim, "Forward kinematics of the general 6–6 Stewart platform using algebraic elimination," *Mechanism Mach. Theory*, vol. 36, no. 9, pp. 1073–1085, Sep. 2001.
[8] J.-P. Merlet, "Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis," *Int. J. Robot. Res.*, vol. 23, no. 3, pp. 221–235, Mar. 2004.
[9] D. Gan, Q. Liao, J. S. Dai, S. Wei, and L. D. Seneviratne, "Forward displacement analysis of the general 6–6 Stewart mechanism using Gröbner bases," *Mechanism Mach. Theory*, vol. 44, no. 9, pp. 1640–1647, Sep. 2009.
[10] G. Liu, Y. Wang, Y. Zhang, and Z. Xie, "Real-time solution of the forward kinematics for a parallel haptic device using a numerical approach based on neural networks," *J. Mech. Sci. Technol.*, vol. 29, no. 6, pp. 2487–2499, Jun. 2015.
[11] M. T. Darvishi and A. Barati, "A third-order Newton-type method to solve systems of nonlinear equations," *Appl. Math. Comput.*, vol. 187, no. 2, pp. 630–635, Apr. 2007.
[12] M. L. N. Gonçalves R. Oliveira, "Convergence of the Gauss–Newton method for a special class of systems of equations under a majorant condition," *Optimization*, vol. 64, no. 3, pp. 577–594, Mar. 2015.
[13] K. Liu, F. L. Lewis, and M. Fitzgerald, "Solution of nonlinear kinematics of a parallel-link constrained Stewart platform manipulator," *Circuits, Syst. Signal Process.*, vol. 13, nos. 2–3, pp. 167–183, Jun. 1994.
[14] P. J. Parikh and S. S. Lam, "Solving the forward kinematics problem in parallel manipulators using an iterative artificial neural network strategy," *Int. J. Adv. Manuf. Technol.*, vol. 40, nos. 5–6, pp. 595–606, Jan. 2009.
[15] A. Morell, M. Tarokh, and L. Acosta, "Solving the forward kinematics problem in parallel robots using support vector regression," *Eng. Appl. Artif. Intell.*, vol. 26, no. 7, pp. 1698–1706, Aug. 2013.
[16] S.-H. Chen and L.-C. Fu, "The forward kinematics of the 6-6 Stewart platform using extra sensors," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Taipei, Taiwan, Oct. 2006, pp. 4671–4676.
[17] V. Petuya, J. Gutiérrez, A. Alonso, O. Altuzarra, and A. Hernández, "A numerical procedure to solve non-linear kinematic problems in spatial mechanisms," *Int. J. Numer. Methods Eng.*, vol. 73, no. 6, pp. 825–843, Feb. 2008.
[18] P. J. Parikh and S. S. Y. Lam, "A hybrid strategy to solve the forward kinematics problem in parallel manipulators," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 18–25, Feb. 2005.
[19] C. Yang, J. He, J. Han, and X. Liu, "Real-time state estimation for spatial six-degree-of-freedom linearly actuated parallel robots," *Mechatronics*, vol. 19, no. 6, pp. 1026–1033, Sep. 2009.
[20] T.-M. Wu, "Solving the nonlinear equations by the Newton-homotopy continuation method with adjustable auxiliary homotopy function," *Appl. Math. Comput.*, vol. 173, no. 1, pp. 383–388, Feb. 2006.
[21] S. M. Varedi, H. M. Daniali, and D. D. Ganji, "Kinematics of an offset 3-UPU translational parallel manipulator by the homotopy continuation method," *Nonlinear Anal., Real World Appl.*, vol. 10, no. 3, pp. 1767–1774, Jun. 2009.
[22] N. Mostashiri, A. Akbarzadeh, and A. Rezaei, "Implementing the homotopy continuation method in a hybrid approach to solve the kinematics problem of spatial parallel robots," *Intell. Service Robot.*, vol. 10, no. 3, pp. 257–270, Jul. 2017.
[23] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Berlin, Germany: Springer, 2011, pp. 52–56.
[24] I. Kardan and A. Akbarzadeh, "An improved hybrid method for forward kinematics analysis of parallel robots," *Adv. Robot.*, vol. 29, no. 6, pp. 401–411, Mar. 2015.
[25] S. Amat, J. A. Ezquerro, and M. A. Hernández-Verón, "On a new family of high-order iterative methods for the matrix pth root," *Numer. Linear Algebra Appl.*, vol. 22, no. 4, pp. 585–595, Aug. 2015.

QIDAN ZHU received the B.S. degree in automatic control and the M.S. and Ph.D. degrees in control theory and control engineering from Harbin Engineering University, Harbin, China, in 1985, 1987, and 2001, respectively. He was a Visiting Scholar with the Kitami Institute of Technology, from 1999 to 2000. He is currently a Professor with the College of Automation, Harbin Engineering University. He has published more than 100 academic articles in national core journals and academic conferences. His research interests include robotics, computer vision, control theory, and engineering.

ZHENG ZHANG is currently pursuing the Ph.D. degree with the School of Automation, Institute of Robotics and Intelligent Control, Harbin Engineering University.

His main research interests include robotics and computer vision.