Kinematic of 4SPRR-SPR parallel robots based on VQTAM

Luo Lan¹,², Hou Li¹, Wu Yang¹, Zhang Qi³ and Wei Yongqiao²

Abstract
4SPRR-SPR parallel robot is a novel closed-loop mechanism. The research of kinematics is the basis of real-time and robust robot control. This paper aims to proposing a method to address a surrogate model of forward kinematics for PMs (Parallel Mechanisms). Herein, the forward kinematics model is derived by training the VQTAM (Vector-Quantified Temporal Associative Memory) network, which originates from a SOM (Self-Organized Mapping). During the processes of training, testing and estimating this neural network, the priority K-means tree search algorithm is utilized, thus improving the training efficacy. Furthermore, LLR (Local Linear Regression), LWR (Local Weighted Linear Regression) and LLE (Local Linear Embedding) algorithms are respectively combined with VQTAM to get three improvement algorithms, aiming to further optimize the prediction accuracy of the networks. To speed up solving the least squared equation in the three algorithms, SVD (Singular Value Decomposition) is introduced. Finally, Data from inverse kinematics by geometric method is obtained, which is for constructing and validating the VQTAM neural network. Results show that the prediction effect of LLE algorithm is better than others, which could be a potential surrogate model to estimate the output of forward kinematics.

Keywords
Parallel robot kinematics, machine learning, VQTAM, priority K-means tree search algorithm, local linearization, SVD

Date received: 23 February 2020; accepted: 7 September 2020
Handling Editor: Lianjun Wu

Introduction
Parallel robots offer noticeable advantages in machining applications, especially in the field of heavy equipment manufacture, due to their high rigid-weight ratio and load capacity. Recently, the machine tool industry has discovered the potential advantages of PMs and many parallel machine tools have been developed based on either the 6-DOF (degrees-of-freedom) parallel mechanisms or 5-DOF PMs.¹ Parallel robots or PMs with 5 or 6 DOF are often used as motion simulators and crane devices. Stewart introduced a 6-DOF PM, which is now popularly known as Stewart platform.² It is commonly used in flight or driving simulators, vibration isolation platform, space docking mechanisms, and space telescope.³,⁴ Hereafter the lower-mobility (2-5 DOFs) PMs have emerged and have become popular owing to their simple structure, low kinematic coupling, low cost and convenient control.⁵–⁷ Xie et al.,⁵ Huang et al.⁶ and Joshi et al.⁷ studied the type synthesis for different types of lower-mobility PM. Therefore the 2UPR/SPR PM⁸ and 3UPS/UP PM⁹,¹⁰ have been used as position adjusters and space telescope (R, P, U and S stand for

¹School of Mechanical Engineering, Sichuan University, Chengdu, China
²School of Mechanical & Electronical Engineering, Lanzhou University of Technology, Lanzhou, China
³School of Intelligent Manufacturing, Panzhihua University, Panzhihua, China

Corresponding author:
Luo Lan, School of Mechanical Engineering, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu, 610065, China.
Email: studyluolan@163.com
revolute, prismatic, universal and spherical joints respectively). As far as 5-DOF PMs are concerned, researchers have mainly worked on the type synthesis. Through the type synthesis, a variety of PMs are obtained. For instance, M. Garcia-Murillo et al. proposed a (3-RRPS) parallel robot and analyzed its workspace. Nicholas Baron, et al. displayed a family of kinematically redundant planar parallel robots, and presented a geometric method of singularity avoidance. In addition, the cable-driven manipulators are a class of PMs, where the end-effector is displaced by means of servo-actuated extendable cables. They have high ratio of payload to robot weight and a much larger workspace which is appropriate for camera positioning in sports stadiums and many others. Arian Bahrami et al. studied the optimal design of various forms of the cable-driven manipulators. In summary, the structures of PMs are various and have been applied to many fields. However, there is still no executable method to solve the kinematics, especially the forward kinematics.

The 5SPRR PM is a 6-DOF PM, which has been used as a machining tool. The structural form of 4SPRR-SPR parallel robot, which has great potential to transform into a new type of mobile machining robot, is derived from the 5SPRR PM. To the best knowledge of the author, up to now, Research, such as its kinematic, workspace analysis et al. has not been conducted yet.

Parallel manipulators do not have formulaic kinematic solutions. Therefore, several studies are focused on methods to simplify the solution of robot kinematics. Intelligent algorithm or machine learning is introduced into kinematics solution of robot. Simpler mapping models of kinematic input and output are established to replace the complex analytical solution. Evolutionary computational methods such as genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) are used to optimize the solutions by formulating the problem in the form of an objective or a fitness function. Furthermore, approximation methods based on artificial neural networks and polynomial regression model can also be used to estimate forward kinematic. Robot kinematics can be regarded as a non-linear autoregressive model with exogenous input (NARX). In reference, the forward kinematics of parallel manipulators is solved by combining the wavelet network and the NARX method. However, the validity of this method is still dependent on parameter initialization and back propagation learning process. The simplification of complex kinematics model often leads to decreasing the accuracy. To overcome this problem, efficient and accurate kinematics identification is needed.

The VQTAM algorithm does not depend on the adjustment of super parameters, but on SOM to achieve topology preservation. It can be applied to the mapping from input space to output space, as an extension of SOM. The mathematical foundation of VQTAM is non-linear manifold embedding. This method is suitable for time series modeling and prediction of non-linear systems. Depending on the self-organizing mapping structure, the discrete mapping relationship can effectively realize output estimation according to exogenous input. Since the input space and output space of VQTAM are discrete, a network with more neurons is required to achieve high-precision prediction. In order to improve the computational efficiency, it is necessary to reduce the number of VQTAM neurons, which decreases the efficiency. Improved local linear algorithms of VQTAM are proposed to solve the inverse kinematics of a 6-DOF robot efficiently and accurately.

In this paper, a novel parallel robot that could be used as mobile machining robot is introduced. The architecture and general kinematic properties of the 4SPRR-SPR parallel robots are outlined, and the study on its kinematic follows.

Research methodology

4SPRR-SPR parallel robot is a novel closed-loop mechanism. Its inverse kinematics can be achieved using a geometric method. The rod length can be obtained quickly through the known pose of end effector. But the forward kinematics of parallel robot is very complex. It’s a non-linear system. Generally, it is difficult to accurately determine the forward kinematics.

Thus, the length of drive limbs corresponding to different poses can be obtained using inverse kinematics conveniently. The data obtained from the inverse kinematics can be used to train the surrogate model of the forward kinematics. In this paper, VQTAM neural network is used to replace the complex calculation of forward kinematics. The research content and structure of this paper are shown in Figure 1.

In Section 2, the inverse kinematics is studied. Using the inverse kinematics algorithm, the length of drive limbs corresponding to poses is calculated as the sample set.

In Section 3, the non-linear system identification method based on VQTAM is studied, and the training method of VQTAM neural network is established. In the process of VQTAM training and testing, it is core to search and activate neuron nodes. Therefore, the priority search K-means tree algorithm is introduced for this step.

In Section 4, the improved method of local linearization of input and output mapping based on VQTAM neural network is studied.
Kinematic of robotic manipulator

4SPRR-SPR parallel robots

The 4SPRR-SPR parallel robot is one of the new 5-DOF PMs, and its structure is shown in Figure 2. The four limbs of the PM consist of one spherical joint, one moving pair and two rotating pairs, and the other one lack for a rotating pair at one end of the limb. The rotating pairs connected with an end effector are coaxial. The spherical joints connected with the static platform are not arranged in the same plane but are distributed in different positions in space. This provides more possibilities for the optimization of the parallel mechanism.

The arrangements of the kinematic pairs of the 4SPRR-SPR parallel robot are similar, except that the end of the first limb is fixed to the moving platform. The kinematic pair arrangement of the $i^{th}$ chain is shown in Figure 3 ($i = 2,3,...,5$).

$A_i$ is the center point of the universal joint connected by the $i$th limb and the static platform, $B_i$ and $C_i$ are the two revolving pair centers of the limb, and $D$ is the reference point of the position of the end effector. The screw of the center line $S$ of each pair is used to represent the motion of the pair. $S_i^1$ and $S_i^2$ are revolting joints; $S_i^3$ is a prismatic joint, and $S_i^4$, $S_i^5$, and $S_i^6$ constitute the rotation of spherical joint in three directions.

For the $i$th limb, the coordinate system $O_{i'}x_{i'}y_{i'}z_{i'}$ at $E_i$ is established. $E_i$ is the vertical foot from $A_i$ to $DC_i$. The screws represented in coordinated $O_{i'}x_{i'}y_{i'}z_{i'}$ are
The screws represented in coordinate system of the end-effector. It can be confirmed that 4SPRR-SPR parallel robots have 5 DOFs. The motion screw system of the first limb differs from those of others, because the end of the first limb is fixed to the moving platform. The screws of the \(i\)th limb constitute the 5-system screws, which provide no constraint to the end effector.

The prismatic joints of the five limbs constitute the drive of a 4SPRR-SPR parallel robot. Therefore, the inverse position solution of the parallel robot can be regarded as the calculation of the rods \(A_iB_i\) about the length \(l_i\) using the coordinates of the known points \(D\) and direction vectors \(S_i\) \((i = 1, 2, \ldots, 5)\). The directions of end effector are the same for the five limbs in unified coordinates.

The geometric condition of the parallel robot is determined using equations (1) and (2).

\[
\begin{align*}
S_i^1 &= (0, 0, 1; 0, 0, 0) \\
S_i^2 &= (0, 1, 0; m_i, 0, c_i) \\
S_i^3 &= (0, 0, 0; n_i, 0, m_i) \\
S_i^4 &= (0, 1, 0; 0, 0, n_i) \\
S_i^5 &= (1, 0, 0; 0, 0, 0) \\
S_i^6 &= (0, 0, 1; 0, -n_i, 0)
\end{align*}
\]

Where \(b_i\), \(c_i\), \(m_i\), and \(n_i\) represent the length of segment \(A_iB_i\), \(B_iC_i\), \(C_iE_i\), and \(E_iA_i\) respectively.

The six screws of the \(i\)th limb are linearly independent \((i = 2, 3, \ldots, 5)\). The screws of the \(i\)th limb constitute a 6-system of screws, which provide no constraint to the end effector.

The reciprocal screw of the 5-system screws can be written as

\[
S_i^a = (0, 1, 0; 0, 0, n_i)
\]

Therefore, five limbs provide one constraint to the end effector. It can be confirmed that 4SPRR-SPR parallel robots have 5 DOFs. The motion screw system which reflects the DOF property of the end-effector can be obtained by finding the secondary inverse screw system of the end-effector.

\[
\begin{align*}
S_1^{cm} &= S_1^r = (1, 0, 0; 0, 0, 0) \\
S_2^{cm} &= S_2^r = (0, 1, 0; 0, 0, 0) \\
S_3^{cm} &= S_3^r = (0, 0, 0; 1, 0, 0) \\
S_4^{cm} &= S_4^r = (0, 0, 0; 0, 0, 1) \\
S_5^{cm} &= S_5^r = (0, 0, 1; 0, -n_i, 0)
\end{align*}
\]

The coordinates of points \(C_i\) and \(B_i\) can be obtained upon calculating \(c_i\). Accordingly the input parameter \(l_i\) corresponding to the pose can be obtained.

The main structural parameters of 4SPRR-SPR parallel robots are the coordinates of points \(A_i\) \((i = 1, 2, \ldots, 5)\) and rod lengths \(c_i\) and \(d_i\). The above parameters can determine the corresponding parallel robot for inverse kinematics calculation.

The inverse kinematics solution of the robot is calculated considering the structural parameters shown in Table 1 as an example.

By determining the position and posture of the moving platform, the coordinate of point \(D\) and the direction vector \(S_i\) is found to be \((0, 1119.17, 3641.27)\), and \((0, 0, -1)\). The length \(l_i\) of the rod \(A_iB_i\), as seen in Table 2, is calculated by substituting these parameters in the aforementioned algorithm.
Table 1. The structural parameters of the parallel robot.

| Parameters | Values (mm) |
|------------|-------------|
| A_1        | (0, 2380.19, 1823.98) |
| A_2        | (-790.54, 1011.96, 1877.15) |
| A_3        | (790.54, 1011.96, 1877.15) |
| A_4        | (-557.90, 2471.63, 2218.87) |
| A_5        | (557.90, 2471.63, 2218.87) |
| d_1        | 440         |
| d_2        | 520         |
| d_3        | 600         |
| d_4        | 1040        |
| d_5        | 1120        |
| c_1        | 180         |

Table 2. The calculations of parallel robot.

| The rod length label | The rod length(mm) |
|----------------------|--------------------|
| l_1                  | 1751.26            |
| l_2                  | 1386.05            |
| l_3                  | 1317.89            |
| l_4                  | 1338.79            |
| l_5                  | 1318.17            |

System identification of non-linear dynamical system using VQTAM

The forward kinematics of the parallel robot aims at solving the corresponding pose given the length of drive limbs. The pose of the end effector is regarded as output $Q$ and the length of the drive limbs as input $p$. The forward kinematics problem of the robot can be transformed into the identification problem of a non-linear dynamic system. The time-discrete difference formula is established as shown in Formula (9).

$$X^{out} = [Q(t + 1)]$$

VQTAM is an extension of SOM, which can be used as an effective manner to realize non-linear mapping. The concept of non-linear manifold embedding can be used to establish the non-linear mapping. SOM is realized through competition and cooperation among feature neurons. Different neurons are activated for different inputs, and the activated neurons affect the output together with the neurons in their neighborhood. Therefore, time series data can be constructed as the training samples of VQTAM.

VQTAM uses the training samples to build a topological structure embedded in multi-dimensional data space, which can construct the mapping relationship between multi-dimensional input vectors and multi-dimensional output vectors. The VQTAM network consists of three layers: input space $Q^{in}$, output space $Q^{out}$ and lattice space. The establishment of the mapping process is shown in Figure 4. The definitions of $X^{in}$ and $X^{out}$ are shown in equations (10) and (11).

$$i^* = \arg \min_{i \in \mathcal{A}} \{ \| X^{in} - \omega_{i}^{in} \| \}$$

$$X^{out} = \omega_{i^*}^{out}$$

Learning strategy of VQTAM

The main parameters of VQTAM are neuron weight vector $\omega_{i}^{in}$, $\omega_{i}^{out}$ and lattice topology. The initial lattice topology structure can be determined as a super parameter before training, so the learning strategy mainly aims at updating the weight vectors $\omega_{i}^{in}$ and $\omega_{i}^{out}$ in the training process.

There is competition and cooperation among neurons in SOM. Different neuron and their neighborhoods are activated by input to participate in the
learning process, and the weight vectors $\omega_1^{in}$ and $\omega_1^{out}$ are updated. The index search of the active neurons in the lattice space is consistent with equation (12).

In order to improve the convergence rate in the learning process, the influence range parameters $\sigma(t)$ of the activation neuron and the learning speed $\alpha(t)$ decrease exponentially with the learning epoch, as shown in equations (14) and (15).

$$\alpha(t) = \alpha_0(\alpha_T/\alpha_0)^{t/T}$$  \hspace{1cm} (14)  

$$\sigma(t) = \sigma_0(\sigma_T/\sigma_0)^{t/T}$$  \hspace{1cm} (15)

Where $t$ is the current learning epoch, and $T$ is the total learning epoch. $\alpha_0$ and $\sigma_0$ are the initial value, and $\alpha_T$ and $\sigma_T$ are the parameter value of training epoch $T$.

The Gauss neighborhood function is used to determine the effect of input on the neighbors of activated neuron, as shown in equation (16).29-41

$$h(i^*, i; t) = \exp\left(-\frac{||i(t) - i^*(t)||^2}{\sigma^2(t)}\right)$$  \hspace{1cm} (16)

$\omega_1^{in}$ and $\omega_1^{out}$ are updated, as shown in equations (17) and (18).

$$\omega_i^{in} \leftarrow \alpha(t)h(i^*, i; t)[X^{in} - \omega_i^{in}]$$  \hspace{1cm} (17)  

$$\omega_i^{out} \leftarrow \alpha(t)h(i^*, i; t)[X^{out} - \omega_i^{out}]$$  \hspace{1cm} (18)

To sum up, the steps involved in the VQTAM algorithm are as follows:

**VQTAM algorithm:**

**(training part)**

Begin

1. Input: $X^{in}$, $X^{out}$ in training set
2. for $j \leftarrow 0$ to length($X^{in}$) - 1
3. $i^* = \arg\min_{i \in A} \{||X_j^{in} - \omega_i^{in}||\}$ /* Search activating neuron according to $X^{in}_{test}$ */
4. $\omega_i^{in} \leftarrow \alpha(t)h(i^*, i; t)[X^{in} - \omega_i^{in}]$ /* Update the weight vector of neuron */

End

**(testing part)**

Begin

1. Input: $X_{test}^{in}$ in testing set
2. for $j \leftarrow 0$ to length($X_{test}^{in}$) - 1
3. $i^* = \arg\min_{i \in A} \{||X_{test_j}^{in} - \omega_i^{in}||\}$ /* Search activating neuron according to $X_{test}^{in}$ */
4. $X_{test}^{out} = \omega_i^{out}$ /* Get the output $X_{test}^{out}$ corresponding to the test data */
5. Output: $X_{test}^{out}$

End

**Searching activated neuron by priority search K-means tree algorithm**

The weight vector $\omega_{i^*}^{in}$ is the nearest neuron to $X^{in}$ in the input space, and is searched during the training, testing and estimation of VQTAM, as shown in equation (12). The search algorithm applied has a great impact on the efficiency of VQTAM. The global traversal method needs to calculate all the Euclidean distance from the weight vectors $\omega_{i}^{in}$ to the input $X^{in}$, and searches the minimum value through traversal, which is very inefficient. The K-Dtree algorithm can reduce the time complexity of searching, but for $d$-dimensional data, the time complexity of K-Dtree data structure search is $O(d^{(d-1)})$. For high-dimensional data search, especially when the data dimension $d>20$, K-Dtree data structure cannot provide high efficiency. $\omega_{i}^{in}$ is 30 dimension. The searching efficiency is severely restricted. The search time complexity of the priority search k-means tree algorithm is $O(Ld \log n/\log K)$, where $n$ is the amount of data in the data set, and $K$ is the number of nearest neighbors to be searched, and $L$
is the maximum number of data retrieved in the search process. The priority search k-means tree algorithm is suitable for searching high-dimensional data.

The priority search K-means tree algorithm is to divide the space into \( B \) different regions using k-means clustering algorithm. Then, the points in each region are partitioned by the same operation until the number of data points in the region is not more than \( K \). In the process of searching, the idea of “divide and conquer” is used to find the nearest point in a smaller area and reduce the amount of data to be searched, so as to improve the efficiency.

The algorithm of establishing a priority search K-means tree data structure is as follows:

**K-means tree data structure building algorithm.**

Begin

1. Input: \( D \rightarrow \omega^i_{in}, B, I_{max} \)
2. /* \( B \) is branch parameter, and \( I_{max} \) is maximum iteration number. */
3. if \( |D| < B \)
4. Create leaf nodes from data sets
5. else
6. \( P \rightarrow \) uses \( C_{alg} \) algorithm to select \( B \) points from data set \( D \)
7. for \( j \rightarrow 0 \) to length(\( P \))
8. do
9. \( C \rightarrow \) clustering data in \( D \) centered on \( P_j \)
10. \( P_j^{new} \rightarrow \) finding the mean value of group \( C \) data after clustering
11. while \( (P = P_j^{new} \& P_j^{new} \) is the non-leaf node)
12. Output: the entire K-means tree data structure

End

**VQTAM local linear improvement algorithms**

The VQTAM algorithm is used to quantify the input space \( \omega^i_{in} \) and the output space \( \omega^o_{out} \). The neurons in \( \omega^i_{in} \) and \( \omega^o_{out} \) correspond through mapping relations. The input \( X^i_{in} \) is approximated to the nearest neuron \( \omega^i_{in} \) during estimation. The accuracy of the estimation results can be guaranteed when the number of neurons is adequate. However, an increase in the number of neurons increases network size and the decreases the computing efficiency. Thus, local linearization of the activated node is used, which can balance the number of neurons and prediction accuracy. Three improved algorithms for VQTAM are proposed: LLR, LWR and LLE based on local linearization. The inverse kinematics of the robot can be further optimized and ensure the computational efficiency.

Local linearization is performed in VQTAM networks. The K-means tree algorithm is used to search the nearest \( n \) data points \( \omega^i_{p_n} \) in the input space and the corresponding output data \( \omega^o_{p_n} \) are mapped, as shown in Figure 5.

**Improvement algorithm of VQTAM with LLR**

It’s assumed that the relationship between local input and output can be expressed linearly, so the LLR can be performed as shown in equation (19).

\[
YB = Z \tag{19}
\]

\( Y \) is the nearest neighborhood \( \omega^i_{p_n} \) of \( X^i_{in} \), and \( Z \) is the corresponding data point \( \omega^o_{p_n} \). As shown in equations (20) and (21).
The linear relationship between the local input and output can be obtained by solving $W$. Equation (21) is an over-determined equation, which is generally treated as a least squares problem and solved by a generalized inverse matrix, as shown in equation (22).

$$B = (Y^T Y)^{-1} Y^T Z$$  \hspace{1cm} (22)

$Y^T Y$ is an $n$-dimensional square matrix. It needs great amount of calculation when the dimension increases. Meanwhile, $Y^T Y$ may be a singular matrix or ill-conditioned matrix, so it is difficult to find the inverse matrix directly. The SVD (singular value decomposition) can be used to solve the least squares problem. $Y$ can be decomposed as shown in equation (23).

$$SVD(Y) = [U][\Sigma][V]^T$$
where $Y \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{l \times l}$  \hspace{1cm} (23)

The column vectors of $U$ are the left singular vectors of $Y$, that is, the eigenvectors of $YY^T$; the column vectors of $V^T$ are the right singular vectors of $Y$, that is, the eigenvectors of $Y^T Y$.

$$T = \begin{bmatrix} \Sigma \\ \theta \end{bmatrix}$$  \hspace{1cm} (24)

Where $\Sigma$ is a diagonal matrix whose value is the singular value $\nu$ of matrix $Y$, that is, the eigenvalue of $YY^T$ is $\lambda = \nu^2$, and $U$ and $V^T$ are orthogonal matrices.

$U$ can be disassembled as $[U_n U_{m-n}]$. For the least squares problem as shown in equation (21), the solution obtained by SVD is shown in Formula (25).

$$B = V \Sigma^{-1} U^T Z$$  \hspace{1cm} (25)

The input $X^{in}$ is used to estimate the output, as shown in Formula (26).

$$\hat{X}^{out} = X^{in} B$$  \hspace{1cm} (26)

**Improvement algorithm of VQTAM with LWR**

The impact of all neighbor points on the prediction results is consistent in LLR algorithm. Considering the different influence of neighbor points, a distance weight is added in the regression process. A local weighted linear regression (LWR) is performed. The cost function is defined as shown in equation (27).

$$J(\varphi) = \sum_{k=1}^{n} w_k \left| \left| \omega^{out}_{ik} - \varphi_k \omega^{in}_{ik} \right| \right|^2$$  \hspace{1cm} (27)

$$w_k = \exp \left( - \frac{\left| \left| X^{in} - \omega^{in}_{ik} \right| \right|^2}{\tau^2} \right)$$  \hspace{1cm} (28)

The analytical solution of the cost function minimization is equivalent to the solution of the over-determined equation shown in equation (29).

$$WY \varphi = WZ$$  \hspace{1cm} (29)

$W$ is an $n$-dimensional diagonal matrix and the diagonal element is $w_k$.

The SVD for matrix $WY$ is also used to solve the over-determined equation.

$$SVD(WY) = [U][\Sigma][V]^T$$  \hspace{1cm} (30)
\[ \varphi = V \Sigma^{-1} U_n^T W Z \]  

**Improvement algorithm of VQTAM with LLE**

It’s assumed that input data can be represented by a linear combination of several samples in its neighborhood. That is to say, the input \( X^{in} \) to be predicted is represented by a linear combination of \( n \) data points in its neighborhood in the input space \( \omega^{in}_{n, \omega} \).

\[
X^{in} = \sum_{k=1}^{n} c_k \omega^{in}_{n,k} \tag{32}
\]

The output has the same linear combination.

\[
X^{out} = \sum_{k=1}^{n} c_k \omega^{out}_{n,k} \tag{33}
\]

\( c_k \) is the coefficients of the linear combination. The output can be estimated using equation (33) after solving \( c_k \). Define the cost function as shown in equation (34) and rewrite it as a matrix.

\[
J(c) = \| X^{in} - \sum_{k=1}^{n} c_k \omega^{in}_{n,k} \|^2 = \| X^{in} - Yc \|^2 \tag{34}
\]

For the least squares problem as shown in equation (33), it can also be solved by SVD of \( Y \).

\[
c = V \Sigma^{-1} U_n^T X^{in} \tag{35}
\]

**Simulation results and discussion**

**Standard VQTAM network test results**

The poses for the sample set are generated according to the robot parameters listed in Table 1, using equations (25) to (30). The generated data can effectively cover the entire workspace of the manipulator.

\[
x(t) = 1500(1 - e^{-\pi t}) \cos 1.88 \pi t \tag{36}
\]

\[
y(t) = 1500(1 - e^{-\pi t}) \sin 1.88 \pi t + 1500 \tag{37}
\]

\[
z(t) = 2000 \cos t + 2000 \tag{38}
\]

\[
\theta_1(t) = \pi \sin t \tag{39}
\]

\[
\theta_2(t) = 2\pi (1 - e^{-\pi t}) \sin 0.86 \pi t \tag{40}
\]

\[ S^i = [\cos \theta_1 \sin \theta_2, \sin \theta_1 \sin \theta_2, \cos \theta_2] \tag{41} \]

where \( t \in [0, 1000] \) and step \( \Delta t = 0.01 \). Thus \( Q(t) = [x(t), y(t), z(t), S^i(t)] \) is calculated as the output of the sample set. The three components of \( S^i \) are \( s_x, s_y \), and \( s_z \), respectively. \( p(t) = [l_x(t), l_y(t), l_z(t), l_0(t), l_3(t)] \) is calculated using the inverse kinematic algorithm. \( p(t) \) and \( Q(t) \) are used as the sample set. Thus the VQTAM network is trained. The data is selected randomly from the sample set in each epoch. At the same time, in order to test the robustness of the algorithm, random errors are added to the test set data according to \( N(0,1) \) distribution.

Table 3 shows the hyper-parameter setting of the VQTAM network training. A VQTAM network of dimension \( 80 \times 80 \) \((M_x \times M_y)\) is trained based on the hyper-parameters, and the effect of training is evaluated using the test set.

Root mean squared error (RMSE), R squared (R\(^2\)), and root mean absolute error (RMAE) are used to evaluate the prediction accuracy of VQTAM for forward kinematics. Table 4 shows the prediction accuracy of the six parameters in output calculated using RMSE, \( R^2 \), and RMAE.

The value of \( R^2 \) is between 0 and 1. The closer it is to 1, the better is the effect of the regression fitting, which means that the learning VQTAM network is more accurate as an approximate model. Generally, \( R^2 > 0.95 \) can be used in engineering applications. For VQTAM networks of dimension \( 80 \times 80 \), the \( R^2 \) values after learning are all above 0.95. This shows that this network is suitable for engineering applications. The RMSE and RMAE reflect the prediction errors in the VQTAM network, and their small values also indicate the prediction accuracy of the network. The convergence result in neural network training process is shown in Figure 6.

It can be seen from Figure 6 that the \( R^2 \) of the VQTAM neural network tested using the test set in the training process gradually increases and converges near 1. This confirms the effectiveness of the VQTAM neural network training algorithm.

The consistency between the actual data and the estimated data can be displayed more intuitively through box diagrams, as shown in Figure 7. The box diagrams of the actual values and predicted values are notably similar, which verifies the excellent prediction ability of the VQTAM network of dimension \( 80 \times 80 \).
VQTAM local linear improvement algorithms

The output of the inverse kinematics problem is estimated using the VQTAM local linear improvement algorithms. The priority search K-means tree search algorithm is used to search the neighbor data of the input. The influence of parameter $k$ on the prediction accuracy is analyzed, considering a VQTAM network with dimension $70 \times 70$ as an example, as shown in Figure 8.

Figure 8 shows that $R^2$ of each parameter increases continuously with the increase in $k$, at first; and it gradually increases to more than 0.95. This indicates the influence on the prediction accuracy. When $k = 7$, the overall prediction accuracy is the best.

**Figure 6.** Convergence result. (a) Prediction accuracy of $x$ in the training process. (b) Prediction accuracy of $y$ in the training process. (c) Prediction accuracy of $z$ in the training process. (d) Prediction accuracy of $s_x$ in the training process. (e) Prediction accuracy of $s_y$ in the training process. (f) Prediction accuracy of $s_z$ in the training process.

**Figure 7.** VQTAM network prediction effect box diagram.
Figure 9 shows that $R^2$ for all variables is more than 0.95 when $k = 3$. Comparing Figure 8 with Figure 9, the overall prediction accuracy of LWR algorithm is higher than that of LLR algorithm. Comparing among Figures 8 to 10, the overall prediction accuracy of improvement algorithm of VQTAM with LLE is the highest. When $k = 7$, the overall prediction accuracy is the best. In conclusion, when the number of neurons is
small, the prediction accuracy of network can be improved by combining local linear algorithm.

**Overall examples and test results**

According to the VQTAM network parameters shown in Table 3, the prediction accuracy of four algorithms under different network dimensions (30 × 30, 40 × 40, 50 × 50, 60 × 60, 70 × 70, 80 × 80) is studied, as shown in Figure 11. The number of the nearest nodes of the local linear algorithm is set to $k = 7$.

Figure 11 shows that an increase of the number of neurons results in a significant increase in the prediction accuracy of the four algorithms. The overall prediction accuracy of improvement algorithm of VQTAM with LLE is the highest. Local linearization of VQTAM yields remarkable results. Using the improved VQTAM algorithm, the prediction accuracy of a network of dimension 50 × 50 is found to be close to that of the standard VQTAM network of dimension 70 × 70.

Considering improvement algorithm of VQTAM with LLE as an example, the output of the forward kinematics is estimated, as shown in Figure 12. The estimated poses are compared with the actual poses. This shows that the VQTAM neural network can effectively estimate the output.

The error of VQTAM can be obtained by comparing the result with the actual pose of the robot. The RSME for position estimation is less than 0.1mm, as shown in Figure 12.
Table 5. This precision has met the control requirements of general robots in industrial applications.

**Conclusion**

The kinematic of 4SPRR-SPR parallel robot is studied, and the inverse kinematics solution by a geometric method is derived. Given the input pose of the robot, this method can be used to calculate the lengths of the drive limbs conveniently and quickly. It can also be used as the computational basis of VQTAM network to establish sample sets.

A fast forward kinematic mapping method for parallel robots using VQTAM for non-linear dynamic systems identification is proposed. The VQTAM algorithm is constructed to train and test the neuron network. Based on the priority K-means tree search algorithm, the training, testing and estimating process of VQTAM network are improved to enhance the search efficiency of nearest neighbors.

According to local neurons activation, the VQTAM network is improved based on local linearization. To further optimize the prediction accuracy of the network, reduce the dimension of the network in application, the improved local linear algorithms for VQTAM networks are proposed.

The sample data set for VQTAM training is constructed. The VQTAM algorithm and its improved algorithm are tested. The test results show that the increase of network dimension can improve the prediction accuracy of VQTAM, but the computational efficiency is affected. By using VQTAM local linearization improvement algorithms, the estimation accuracy can be optimized for low-dimensional network. The above algorithm can show good prediction accuracy, and the prediction effect of LLE algorithm is the best among the three algorithms.

**Acknowledgements**

The authors would like to acknowledge the financial support.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project is supported by National Natural Science Foundation of China (Grant No. 5187052280) and by Economic and Information Commission of Sichuan Province (Development of Normal-position Intelligent Manufacturing Technology and Device for Major Equipment).

**ORCID iD**

Luo Lan https://orcid.org/0000-0002-2617-7774

**References**

1. Gao F, Peng B, Zhao H, et al. A novel 5-DOF fully parallel kinematic machine tool. *Int J Adv Manuf Technol* 2006; 31(1): 201–207.
2. Stewart D. A platform with six degrees of freedom. *P I Mech Eng* 1965; 180: 371–386.
3. Chen X, Liu XJ, Xie FG, et al. A comparison study on motion/force transmissibility of two typical 3-DOF parallel manipulators: the sprint Z3 and A3 tool heads. *Int J Adv Robot Syst* 2014; 11: 11–19.
4. Bahrami A, Tafaoli-Masoule M and Bahrami M. Active vibration control of piezoelectric Stewart platform based on fuzzy control. *Int J Mat Mech Eng* 2013; 2: 17–22.
5. Xie H, Li S, Shen YF, et al. Structural synthesis for a lower-mobility parallel kinematic machine with swivel hinges. *Robot Comput Integr Manuf* 2014; 30: 413–420.
6. Huang Z and Li QC. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method. *Int J Rob Res* 2003; 22: 59–79.
7. Joshi SA and Tsai LW. The kinematics of a class of 3-dof, 4-legged parallel manipulators. *J Mech Des* 2003; 125: 52–60.
8. Jin Y, Bi ZM, Liu HT, et al. Kinematic analysis and dimensional synthesis of exechon parallel kinematic machine for large volume machining. *J Mech Rob* 2015; 7: 041004.
9. Siciliano B. The Tricept robot: inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm. *Robotica* 1999; 17: 437–445.
10. Sheng X, Xu L and Wang Z. A position-based explicit force control strategy based on online trajectory prediction. *Int J Rob Auto* 2017; 32: 93–100.
11. Fang Y and Tsai LW. Structure synthesis of a class of 4-DoF and 5-DoF parallel manipulators with identical limb structures. *Int J Robot Res* 2002; 21: 799–810.
12. Huang Z and Li QC. General methodology for type synthesis of symmetrical lower-mobility parallel manipulators and several novel manipulators. *Int J Robot Res* 2002; 21: 131–145.
13. Kong X and Gosselin C. Type synthesis of 5-DOF parallel manipulators based on screw theory. *J Robot Syst* 2005; 22: 535–547.
14. Kong X and Gosselin C. *Type synthesis of parallel mechanisms*. Heidelberg: Springer, 2007, 33.
15. Zhu SJ and Huang Z. Eighteen fully symmetrical 5-DoF 3R2T parallel manipulators with better actuating modes. *Int J Adv Manuf Technol* 2007; 34: 406–412.
16. Wang L, Fang Y, Qu H, et al. Design and analysis of novel 2R1T generalized parallel mechanisms with large rotational angles. *Mech Mach Theory* 2020; 150: 103879.
17. Xu Y, Zhao Y, Yue Y, et al. Type synthesis of over-constrained 2R1T parallel mechanisms with the fewest kinematic joints based on the ultimate constraint wrenches. *Mech Mach Theory* 2020; 147: 103766.
18. Garcia-Murillo M, Nunez-Altimirano D, Gallardo-Alvarado J, et al. Algorithm to determine the volume
and shape of constant orientation workspace for a 2(3-RRPS) parallel robot. *IEEE Lat Am Trans* 2020; 18: 1156–1163.

19. Baron N, Philippides A and Rojas N. A robust geometric method of singularity avoidance for kinematically redundant planar parallel robot manipulators. *Mech Mach Theory* 2020; 151: 103863.

20. Idá E, Marian D and Carricato M. A deployable cable-driven parallel robot with large rotational capabilities for laser-scanning applications. *IEEE Robot Autom Lett* 2020; 5: 4140–4147.

21. Li L, Fang Y and Wang L. Design optimization of a 3-D three cable driven manipulator. *Mech Mach Theory* 2020; 148: 103802.

22. Bahrami A and Nikkhah Bahrami M. Optimal design of a spatial four cable driven parallel manipulator. In: *Proceedings of the 2011 IEEE international conference on robotics and biomimetics*, Karon Beach, Phuket, Thailand, 7–11 December 2011, pp. 2143–2149. IEEE.

23. Bahrami A and Nikkhah-Bahrami M. Optimal design of a spatial six-cable robot. In: *Proceedings of the IASTED international conference*, Pittsburg, 7–9 November 2011, pp. 134–141. USA: ACTA Press.

24. Bar GF and Weib G. Kinematic analysis of a Pentapod robot. *J Geometry Graph* 2006; 10: 173–182.

25. Alagheband A, Mahmoodi M, Mills JK, et al. Comparative analysis of a redundant Pentapod parallel kinematic machine. *J Mech Rob* 2006; 7/034502: 1–7.

26. Borras J and Thomas F. Singularity invariant leg substitution in Pentapod. In: *The 2010 IEEE/RSJ international conference on intelligent robots and systems*, Taipei, 18–22 October 2010, 978-1-4244-6676-4/10, pp. 2766–2771. TAIWAN: IEEE.

27. Boudreau R and Turkkan N. Solving the forward kinematics of parallel manipulators with a genetic algorithm. *J Robot Syst* 1996; 13: 111–125.

28. Sheng L, Wan-long L, Yan-chun D, et al. Forward kinematics of the Stewart platform using hybrid immune genetic algorithm. In: *2006 international conference on mechatronics and automation*, Henan Univ Sci & Technol, Luoyang, 25–28 June 2006, pp. 2330–2335. People’s Republic of China: IEEE.

29. Rolland L and Chandra R. Forward kinematics of the 6-general parallel manipulator using real coded genetic algorithms. In: *2009 IEEE/ASME international conference on advanced intelligent mechatronics*, Singapore, 14–17 July 2009, pp. 1637–1642. Singapore: IEEE.

30. Omran A, Bayoumi M, Kassem A, et al. Optimal forward kinematics modeling of a Stewart manipulator using genetic algorithms. *Jordan J Mech Ind Eng* 2009; 3: 280–293.

31. Zhang D and Gao Z. Forward kinematics, performance analysis, and multi-objective optimization of a bio-inspired parallel manipulator. *Robot Com Int Manuf* 2012; 28: 484–492.

32. Wang X-S, Hao M-L and Cheng Y-H. On the use of differential evolution for forward kinematics of parallel manipulators. *Appl Math Com* 2008; 205: 760–769.

33. Akbas A. Application of neural networks to modeling and control of parallel manipulators. In: Ryu J-H (ed.) *Parallel manipulators, new developments*. Vienna Austria: In Tech., 2008, pp. 21–40.

34. Tarokh M. Real time forward kinematics solutions for general Stewart platforms. In: *Proceedings 2007 IEEE international conference on robotics and automation*, Roma, Italy, 10–14 April 2007, pp. 901–906. InTech.

35. Morell A, Acosta L and Toledo J. An artificial intelligence approach to forward kinematics of Stewart platforms. In: *20th Mediterranean conference on control and automation (MED)*, Barcelona, Spain, 3–4 July 2012, pp. 433–438. IEEE.

36. Yee Cs and Lim K-b. Forward kinematics solution of Stewart platform using neural networks. *Neurocomputing* 1997; 16: 333–349.

37. Lee Hyung S and Myung-Chul H. The estimation for forward kinematic solution of Stewart platform using the neural network. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Kyongju, South Korea, 17–21 October 1999, 501, pp. 501–506. IEEE.

38. Kumar PR and Bandyopadhyay B. The forward kinematic modeling of a Stewart platform using NLRAX model with wavelet network. In: *IEEE international conference on industrial informatics*, Bochum, Germany, 29–31 July 2013, pp. 343–348. IEEE.

39. Limtrakul S and Arnonkijpanich B. Supervised learning based on the self-organizing maps for forward kinematic modeling of Stewart platform. *Neural Comput Appl* 2019;31: 619–635.

40. Heskes T. Self-organizing maps, vector quantization, and mixture modeling. *IEEE Trans Neural Netw* 2001; 12: 1299–1305.

41. Barreto GD and Araujo AFR. Temporal associative memory and function approximation with the self-organizing map. In: *Neural networks for signal processing XII, proceedings*, 2002, pp. 109–118.

42. Muja M and Lowe DG. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans Pattern Anal Mach Intell* 2014; 36: 2227–2240.