

Design and Implementation of a 6-DOF Intelligent Single-arm Robot

Feng Chen¹, Chao Lu^{1*}, Yunhong Ma¹, Xuesi Li², Xiaojin Chai¹ and Shixiong Wang¹

¹School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China

²China Academy of Launch Vehicle Technology, Beijing 100076, China

*Corresponding author

Abstract—The design and implementation of single-arm robots are crucial offshoots of modern industry. It is of great significance that the device mentioned above could replace people to complete a great deal of complicated and dangerous industrial operations. Instead of the analysis and the introduction of the mechanical structure of designing a robotic manipulator itself, the existing literatures mostly pay their attention on the kinematics and dynamics modeling and control problems of the robots, giving no essential usefulness to the serious of problems in designing and actualizing the specific-function-oriented robots in special work environment or for particular operating purpose. In addition, the traditional methods proposed before of handling forward kinematics and inverse kinematics of those robots are either complex in modeling or hard to ensure the ideal real-time performance. To this end, by elaborately analyzing the characteristics of geometric model of single-arm-and-multi-DOF cascaded robotic manipulator, the paper introduces the design principles of the robots in different using background and proposes an intelligent real-time scheme of calculating forward and inverse kinematics based on neural network, through which we provide readers with complete instructive guidelines in designing and implementing a single-arm robot.

Keywords- single-arm robot; mechanical design; system design

I. INTRODUCTION

Robotics integrating computer science, cybernetics, mechanics, information and sensing technology, artificial intelligence, bionics and so on, is a fascinatingly new and impressively high technology as well as a growing industry. Technically, the essence of Robotics is the combination of four technologies including perception, decision-making, action-selecting and interaction. It has been no wonder an increasingly-widely-applied-in-practice research field attracting more and more scholars [1-2]. The robotic manipulator, which could grip and move work pieces automatically in the automatic production process, is a kind of newborn device developed accordingly with the rhythm of mechanization and automatization, and is also an important part of modern industry significantly freeing people from completing the complicated and dangerous work in some extreme conditions.

However, literatures previous mostly pay their attention more on the kinematics and dynamics modeling and control problems of the robots than on the analysis and design of the mechanical structure of the robotic manipulator itself [1-2,4-6], providing for the serious of problems in designing and implementing the specific-function-oriented robots in special

work environment or for particular operating purpose limited help. Besides, methods formerly presented of handling forward kinematics and inverse kinematics of those robots have generally two distinct branches [7-9]. The first category is the so-called analytical method requesting rigorously completely analytical formularization and a perplexing and intricate process. The alternative one is the method of numerical iteration which asks digital computers to iteratively work out the solution of kinematics both forward and inverse in limited time duly and which could not effectively ensure the real-time performance of the robotic systems.

Aiming at the problems aforementioned, the paper elaborates the characteristics of geometric model of single-arm-and-multi-DOF cascaded robotic manipulator, puts forward the design principles of the robots in different using background and proposes an intelligent real-time scheme of calculating forward and inverse kinematics based on neural network, through which we provide readers with complete instructive guidelines in designing and implementing a single-arm robot.

The organization of the work is as follows. In the second part, we firstly present some basic mathematical theories providing theoretical grounds for the design of a manipulator and, further, the principles and guiding schemes of designing a single-arm robot (robotic manipulator) will be discussed in the third chapter. Eventually, the fourth section will study the intelligent real-time algorithm of kinematics, both forward and inverse, based on neural network.

II. PRELIMINARIES

A. The Establishment of Moving Coordinate and the Theory of Homogeneous Transformation

As shown in Figure 1, the side view of a manipulator is given, in which the world coordinate Σ_w , also known as the global coordinate (marked in green), is fixed on the substrate being given its axes x , y and z , and the arm coordinates, Σ_a , Σ_b ..., also known as the local coordinates (marked in red), are designated as the dynamic coordinates. Figure 1 (A) shows the initial setting that the world coordinate Σ_w has consistent axial orientations with the arm coordinates, Σ_a , Σ_b . Inasmuch as the substrate is fixed, the world coordinate will keep its states constant, rotatorily and translationally. The transforming relation of each coordinate are illustrated in Table I.

From Figure 1 (A), the relation between the position, aP_b , of the second joint b of the manipulator in the Σa and that, P_b , of the origin in the world coordinate could be given as (the position of the first joint a in the world coordinate, p_a , coincides with P_o , the origin of the world coordinate):

TABLE I. THE TRANSFORMING RELATION OF EACH COORDINATE

| Origin | Mother Coord. | Position in its Mother Coord. | Rotation around its Mother Coord. | | | Position in World Coord. |
|----------|---------------|-------------------------------|-----------------------------------|----------|-----------|--------------------------|
| | | | x | y | z | |
| <i>o</i> | / | / | / | / | / | $[0,0,0]$ |
| <i>a</i> | <i>o</i> | oP_a | 0 | θ | 0 | P_a |
| <i>b</i> | <i>a</i> | aP_b | 0 | 0 | α | P_b |
| <i>c</i> | <i>b</i> | bP_c | 0 | 0 | β | P_c |
| <i>d</i> | <i>c</i> | cP_d | γ | 0 | φ | P_d |

$$p_b = p_a + {}^a p_b \quad (1)$$

where p_a, p_b are all 3-D vectors. Taking p_b for an example, it is represented as:

$$p_b = \begin{bmatrix} p_{bx} \\ p_{by} \\ p_{bz} \end{bmatrix}. \quad (2)$$

Plus, when the manipulator is, somehow, outstretched, as Figure 1 (B) illustrated, we can, in return, get the position of the joint b

$$p_b = p_a + {}^a p'_b \quad (3)$$

where ${}^a p'_b$ is the result of rotation transformation of ${}^a p_b$. Specifically, it is given by rotating the angle of α around the axis z of its mother coordinate, Σa . Thus the base vectors of the transformation can be obtained:

$$e_{ax} = \begin{bmatrix} \cos \alpha \\ \sin \alpha \\ 0 \end{bmatrix} \quad e_{ay} = \begin{bmatrix} -\sin \alpha \\ \cos \alpha \\ 0 \end{bmatrix} \quad e_{az} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (4)$$

The rotation matrix, R_a , determined by the three base vectors is consequently drawn as follows:

$$R_a \equiv [e_{ax} \quad e_{ay} \quad e_{az}]. \quad (5)$$

Namely

$$R_a = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

Using the matrix above, the relation between ${}^a p'_b$ and ${}^a p_b$ could then be given as:

$${}^a p'_b = R_a {}^a p_b. \quad (7)$$

Thus

$$p_b = p_a + R_a {}^a p_b. \quad (8)$$

When (8) is rewrote into the matrix form, we can get:

$$\begin{bmatrix} p_b \\ 1 \end{bmatrix} = \begin{bmatrix} R_a & p_a \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^a p_b \\ 1 \end{bmatrix}. \quad (9)$$

If we set

$$T_a = \begin{bmatrix} R_a & p_a \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

Then T_a could always be referred to as Homogeneous Transformation Matrix (HTM). Actually, the HTMs bridge the representation of a point in the local coordinate with that in its mother coordinate or in the world coordinate. Thus, the HTMs essentially describe the position and the posture of the arms.

As for the local coordinate Σc , its HTM is accordingly given as T_c^a , which showed in (11)

$$T_c^a = \begin{bmatrix} R_c^a & {}^a p_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where ${}^a p_c$ is the representation of the position of a point, which is within the Σc , in Σa and T_c^a is the HTM from Σc to Σa .

Suppose there is a multi-joint cascaded manipulator simultaneously link the coordinates fixed in joints from $\Sigma 1$ to ΣN , and the HTM between the two adjacent coordinates Σi and $\Sigma i+1$ is T_{i+1}^i , then we can further know that

$$T_N = T_1^2 T_2^3 \dots T_N^{N-1}. \quad (12)$$

T_N in (12) is the HTM between the Nth joint and the world coordinate which is its base coordinate.

If a new joint were to be added in the free end of the manipulator, the corresponding HTM should be multiplied in

the right of the T_N . The law addressed just is termed The Theory of Homogeneous Transformation.

B. The theorem of determining a quadrilateral

Theorem 1: For the quadrilateral ABCD, shown in Figure 2, known that A and B are two fixed points invariably, C and D are two movable points, and line-segments AB, BC, and CD are constant in length. If the position of point D were to be fixed at a certain case, the position of the point C will be

uniquely determined, as a result. That is, the values of and will be uniquely determined.

Proof: Connect the two points B and D to generate an auxiliary dotted line BD, after which the quadrilateral is divided into two independent triangles, ABD and BCD. For the triangle ABD is pre-determined, the triangle ABD is, thus, uniquely determined. Hence, in the triangle BCD, the three edges are, thereafter, determined together with the geometric-shape of BCD and the values of and. Finally, since the B and the D are fixed, the position of the point C is uniquely given.

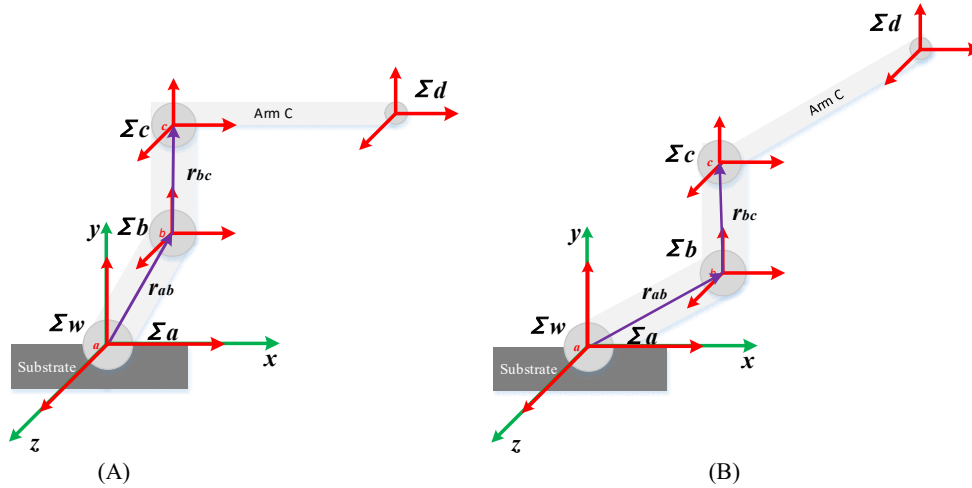


FIGURE I. THE WORLD COORDINATE AND LOCAL MOVING COORDINATES FIXED ON ARMS: (A) INITIAL STATE, (B) THE STATE AFTER THAT THE JOINTS, A AND C, ROTATED.

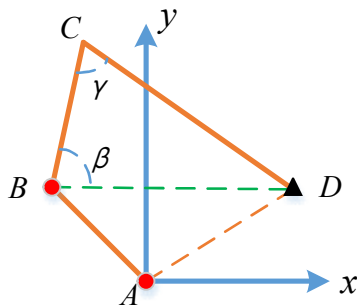


FIGURE II. DIAGRAM OF THE QUADRILATERAL ABCD.

C. The Forward and Inverse Kinematics

The spatial state of each joint of the manipulator depends exclusively on the positions and the postures of their mother joints. The process to determine angle of each joint with specific model knowing the positions and postures of all joints is termed Inverse Kinematics, while the process using angles of all joints inversely calculates the position and posture of each joint is referred to as forward kinematics. Apparently, the two processes are, to each other, inverse [10].

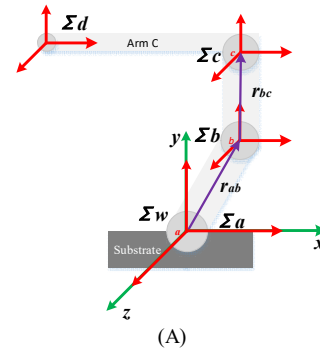
To complete the intended actions, the hand of the manipulator must have appropriate position and posture, which is determined by proper angle of every joint. Therefore, the analysis of kinematics both forward and inverse is necessary. It

is also the core purpose of setting the driving model, kinematically.

III. THE PRINCIPLES AND GUIDING SCHEME OF DESIGNING A SINGLE-ARM ROBOT

A. Purpose-driven Design Principles

Due to the different applications, the structure of manipulator varies in different forms, so do the driving mode



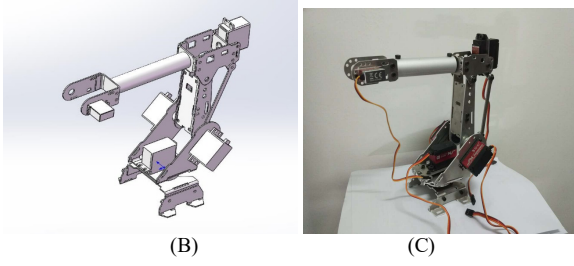


FIGURE III. MODELS OF A GENERALLY-USED 6-DOF SINGLE-ARM ROBOT: (A) GEOMETRICAL MODEL, (B) MECHANICAL MODEL, (C) PHYSICAL MODEL.

of each part as well as the transmission mechanism. Usually, according to the function of various parts of the robot, the mechanical frame (body) of the robot mainly consists of the following sections:

1) **Hand.** An operating organ arranged on the end of the wrist for the purpose of carrying out operations, sometimes referred to as gripper or end-effector. It generally is a rotary or translational type (mostly is a rotary type, for its simple structure) and is no more than two fingers (grripper, in some special cases, multi-finger is also reasonable). What's more, according to the need of grasping, it can be classed into two kinds working inward or outward. Also, it could be negative pressure sucker, vacuum sucker (which is mainly used to hold the cold or smooth-surface sheet or metal parts) or electromagnetic sucker.

2) **Wrist.** The part that bridges the hand and the adjacent arm and that is used primarily to change the spatial position and posture of the hand and to transfer the load to the arm. It usually has three types of motion: up-down (elevation), left-right (azimuth), and rotation around its own axis. With special requirements, the wrist can also be provided with a small-distance lateral shift.

3) **Arms.** The parts of mechanically linking the substrate and the wrist, including the big arm, the small arm generally, and other arms particularly (see Figure 1 (B)). The main role of those is to change the spatial position and posture of the hand, further to meet the accessible-space request of the manipulator, and to transfer loads to the substrate. It generally has three degrees of freedom, which can be represented in the Cartesian coordinate, cylindrical coordinate, spherical coordinate and even multi-joint cascaded system (in which case that the arm can flex randomly).

4) **Substrate.** The basis part belonging to the single-arm robot, holding the function of support, always being fixed on a stationary or moving platform, accordingly.

The main structure of the robot consists of four all parts listed above. In usual the manipulator aims to act the demanded movements within a requested space range with a certain load and at a certain speed. Therefore, the following principles should be followed in carrying out the design:

1) The axis of joints should be parallel to each other as much as possible, while those orthogonal joints should be deployed at a same origin point as much as it can, so that the

kinematics analyses and calculation could be simplified, which is beneficial to modeling and control.

2) The structure and size of the robot should be reasonable to meet the requirement of working space and operating purpose.

3) In order to improve the precision of motion and its control of the manipulator, the robot should be manufactured with light-weight materials that yet can provide sufficient strength and rigidity to reduce the weight of it. Meantime, the structural optimization should also be considered in design to further limit the weight, if possible.

4) The bearing clearance (gap) of each joint of the manipulator should be reduced as much as possible, so as to reduce the movement error caused by it. Thereby, all joints should be configured with reliably-worked and easy-to-operate adjustment mechanism.

5) The movement of the joints of the manipulator should be restricted within a limited range using position-confining sub-modules, structurally. Also, space for installing transmission or driving parts (actuator or steering engine) and cables should be planned and pre-reserved.

B. The Guiding Design Scheme of a Single-arm Robot

According to the design principles and the general using requirements of the manipulator, we choose the multi-joint 6-DOF structure as our mechanical model. The multi-joint model has kinds of interesting traits and qualities like compactness, charming accuracy in positioning, and small volume. The DOFs of the single-arm robot are respectively assigned as hand open-close, wrist rotation, wrist elevation, small-arm elevation, big-arm elevation, and substrate rotation, as shown in Figure 3. Meanwhile, its geometric parameters are provided in Table II.

TABLE II. GEOMETRIC PARAMETERS OF THE ROBOT MODEL

| Mechanical Parts | Description of the parameter | Value of the parameter |
|---|--|-------------------------------|
| Joint a | Changing Range of θ around y | $0^\circ \sim 360^\circ$ |
| Joint b | Changing Range of α around z | $40^\circ \sim -90^\circ$ |
| Joint c | Changing Range of β around z | $40^\circ \sim -30^\circ$ |
| Linking arm between Joint a and Joint b | Spatial Vector r_{ab} (aP_b) | $[4.5e-2, 4.0e-2, -2.5e-3] m$ |
| Linking arm between Joint b and Joint c | Spatial Vector r_{bc} (bP_c) | $[0, 16.0e-2, 0] m$ |
| Linking arm between Joint c and Joint d | Spatial Vector r_{cd} (cP_d) | $[-20.0e-2, 2.0e-2, 0] m$ |

IV. NN-BASED INTELLIGENT REAL-TIME ALGORITHM OF FORWARD AND INVERSE KINEMATICS

Enlightened by the discussions in Section II-A, we know that the position of the joint d, P_d , is

$$P_d = P_c + {}^cP'_d = P_c + R_c \cdot {}^cP_d. \quad (13)$$

Likewise,

$$P_c = P_b + {}^b P'_c = P_b + R_b \cdot {}^b P_c, \quad (14)$$

$$P_b = P_a + {}^a P'_b = P_a + R_a \cdot {}^a P_b, \quad (15)$$

$$P_a = P_o = [0, 0, 0]^T. \quad (16)$$

Substituting some parts of (13) with (14)~(16), we can then get

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = P_d = Y_\theta^T Z_\alpha^T Z_\beta^T {}^c P_d + Y_\theta^T Z_\alpha^T {}^b P_c + Y_\theta^T {}^a P_b \quad (17)$$

where Y_θ^T is the rotation matrix of θ running around the axis y in the world coordinate, T denotes the transpose of a matrix; Z_α^T is the rotation matrix of α running around the axis z in the world coordinate and T indicates the transpose of a matrix, similarly; Parameters left can be found in Table II.

At this case, (17) clearly is the calculating pattern of the forward kinematics. If it should be mathematically reversible, that is, the mapping from $P_d = [x_d, y_d, z_d]$ to $[\theta, \alpha, \beta]$ should be exist, then the inverse mapping of (17) would be the calculating pattern of the inverse kinematics.

Evidently, the four joints, symbolized by four points, a, b, c, d, in Figure 2, together with arms linking with them, construct a quadrilateral. In the light of Theorem 1, with a further step, it is definite that the mapping from $P_d = [x_d, y_d, z_d]$ to $[\theta, \alpha, \beta]$ indeed exists, which means that the solution of inverse kinematics could be directly achieved from (17).

Nonetheless, (17) has a strong characteristic of non-linearity. Therefore, the analytic solution of the inverse problem is unattainable. Besides, in the consideration of that the numerical calculating methods are usually real-time deficiency, we adopt a NN-based approach holding the traits of intelligence in some sense to fit the kinematical models, both forward and inverse, for simplicity and the requirement of real-time. The algorithm is showed below.

Algorithm 1: NN-based intelligent real-time algorithm of forward and inverse kinematics.

Step 1:Generate the training data using (17). The changing range of the three degrees $[\theta, \alpha, \beta]$ could be retrieved from Table 2. Setting the step size of each degree as 1 degree, then $P_d = [x_d, y_d, z_d]$ could be calculated by (17);

Step 2:Train a Neural Network (NN) to be the calculating model of forward kinematics using the data provided by Step 1, from $[\theta, \alpha, \beta]$ to $P_d = [x_d, y_d, z_d]$;

Step 3: Train a NN to be the calculating model of inverse kinematics using the data provided by Step 1, from $P_d = [x_d, y_d, z_d]$ to $[\theta, \alpha, \beta]$;

Step 4: End of the NN-based kinematics calculating algorithm.

The analysis above shows, obviously, that the computational efficiency of handling the forward and inverse kinematics based on neural network will depend greatly on the selection of network types and their related network parameters. However, the structure of data to be processed is ideal (pretty good), the noise is not dominant, and the training-sample set is large enough. Thus, the usual NNs have ideal performances to be applied, which could free us from designing a specific one.

A. Kinematic Solving Method Based on BP Neural Networks

That arbitrary nonlinearity is sufficient to represent all functions by neural networks, which reflects the powerful ability of NN, has been proposed in [11]. In this section, two BP neural networks are applied to fit forward and inverse kinematics solving models respectively. In order to investigate the performance of BP nets, different hidden-layer structures are considered. The training set of the networks consists of the samples from the kinematics models (see Algorithm 1). It is important to note that the test set, for the purpose of efficiently validating the performance of the net, is completely different from the training set.

The experiment is divided into four groups, which explore the BP nets with single hidden-layer and double hidden-layer comprised of different neurons, respectively. The detailed experimental results are shown in Figures 4-7.

Figures 4-7(A) show that the training errors gradually decrease with the training process goes on. After 100 epochs, the curve of training error tends to be smooth, which indicates the convergence. Meanwhile, if the number of hidden layers are same, the more the number of neurons is, the stronger the ability to fit the data is. Figure 4-7(B) and (C) show the same views obtained by the neural networks and the actual data, respectively. Since the data points for fitting has three dimensions, the X-Y view is selected here for display. Similarly, under the condition that the network layers are same to each other, if the number of neurons increases, the output of the networks is closer to the real data. In addition, in contrast to Figure 4-6, it is obvious that more hidden layers do not necessarily improve the performance of nets, unless there are enough neurons included in the net, as showed in Figure 7.

B. Kinematic Solving Method Based on RBF Neural Networks

The RBF neural network, which belongs to the class of the forward neural network, can approximate any nonlinear function [12]. Moreover, the RBF net has good generalization ability and fast learning convergence speed. In this section, we

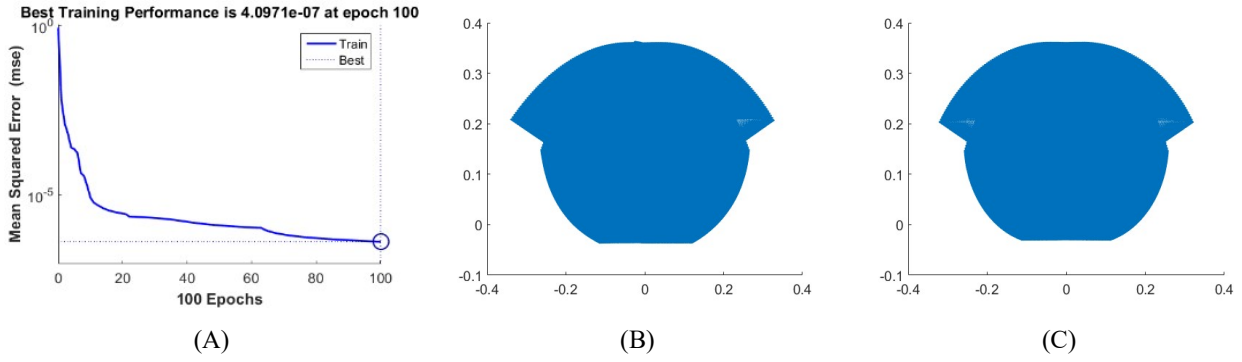


FIGURE IV. THE RESULTS OF BP NEURAL NETWORKS TRAINED WITH STRUCTURE [20] : (A) THE CURVE OF TRAINING ERROR(B) THE X-Y VIEW OF FITTING RESULT(C) THE X-Y VIEW OF GROUND TRUTH

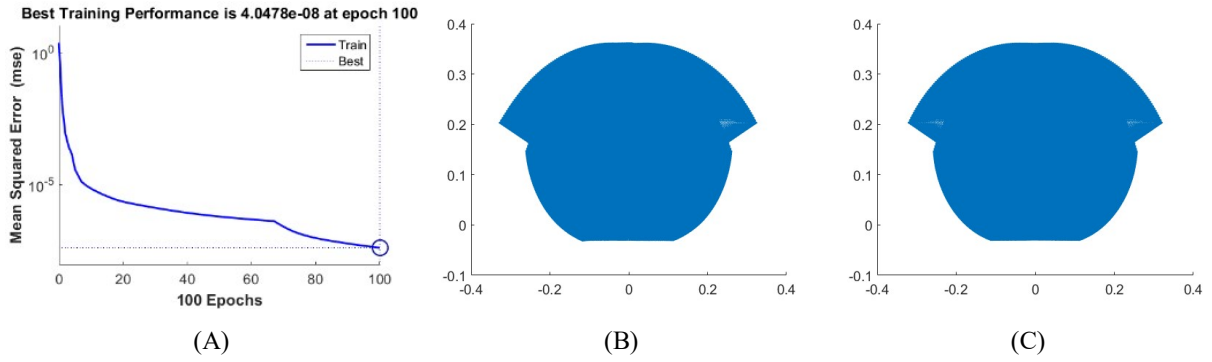


FIGURE V. THE RESULTS OF BP NEURAL NETWORKS TRAINED WITH STRUCTURE [50] : (A) THE CURVE OF TRAINING ERROR(B) THE X-Y VIEW OF FITTING RESULT(C) THE X-Y VIEW OF GROUND TRUTH

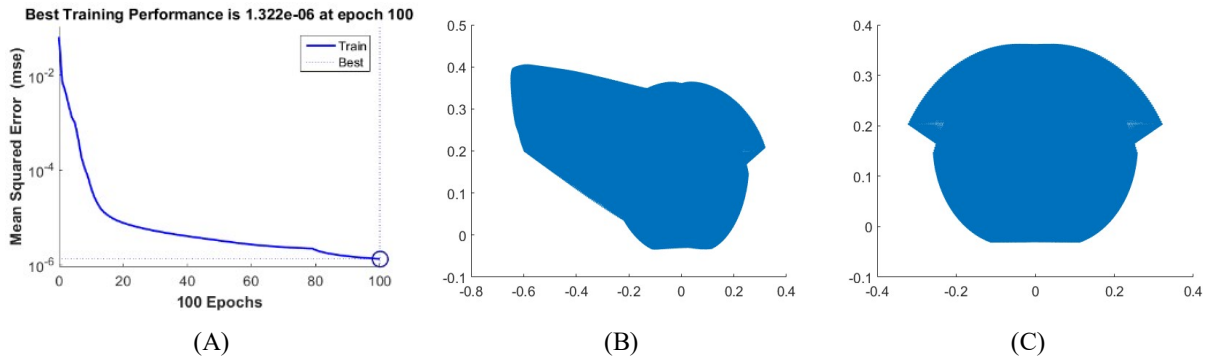


FIGURE VI. THE RESULTS OF BP NEURAL NETWORKS TRAINED WITH STRUCTURE [10 5] : (A) THE CURVE OF TRAINING ERROR(B) THE X-Y VIEW OF FITTING RESULT(C) THE X-Y VIEW OF GROUND TRUTH

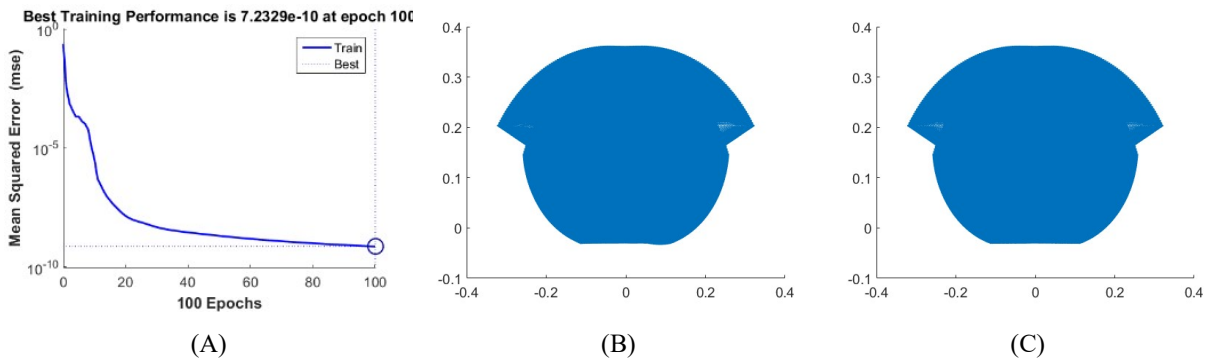


FIGURE VII. THE RESULTS OF BP NEURAL NETWORKS TRAINED WITH STRUCTURE [50 10] : (A) THE CURVE OF TRAINING ERROR(B) THE X-Y VIEW OF FITTING RESULT(C) THE X-Y VIEW OF GROUND TRUTH

utilize the RBF neural network, which can automatically increase the number of neurons in the hidden layer until the mean variance of model meets the precision requirement, to fit the given data. Therefore, we only explore the generalization ability of RBF networks with different number of training samples.

When the number of training samples is 100 and 300, respectively, the experimental results are shown in Figure 8-9. As seen in Figure 8, although there are only 100 training samples, the network has a very strong fitting capability, and the results from the X-Z view are a little different from the actual data. However, when the number of training sample is

300, the results from the X-Z view are basically consistent with the actual data.

C. *Selecting Criteria of From the Results of the Experiments, We Could Get the Fact That BP Net and RBF Net both Have Strong Data Fitting Ability for This Kinematics Problem.*

According to the actual performance, the following selection criteria, which should be considered in selecting the type of a neural network and its proper parameters, are summarized:

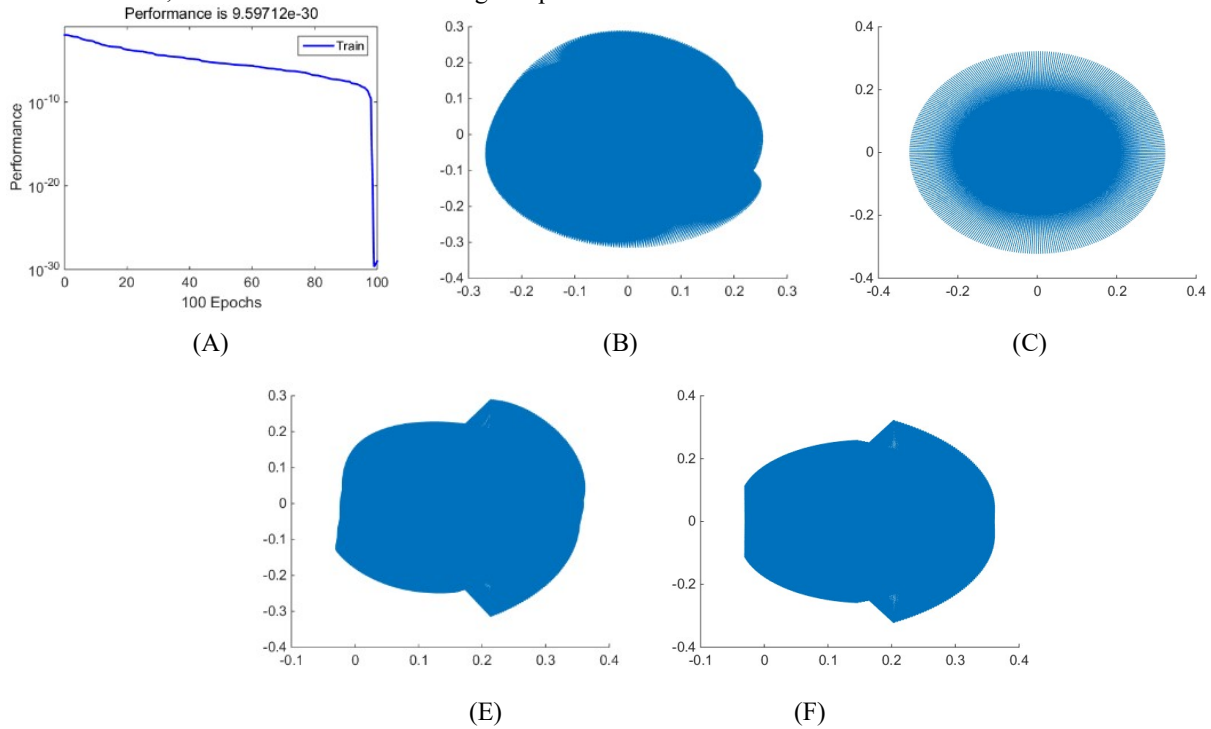
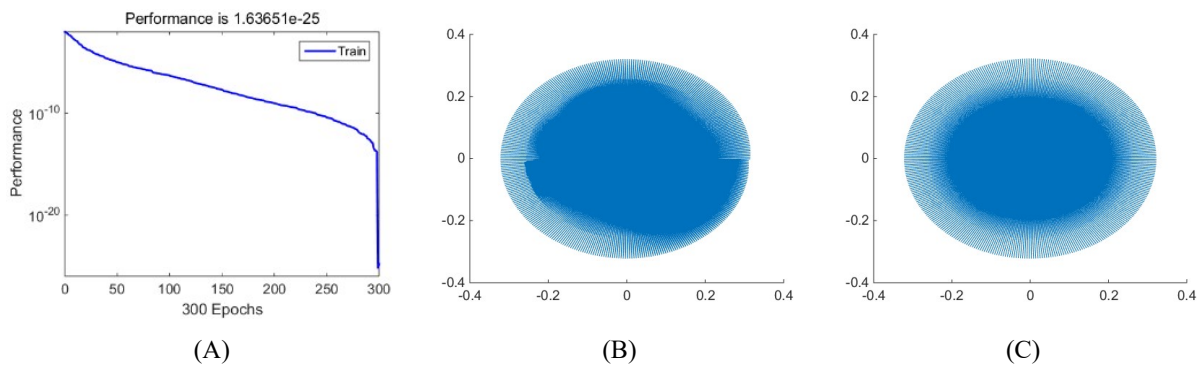


FIGURE VIII. THE RESULTS OF RBF NEURAL NETWORKS TRAINED WITH 100 SAMPLES : (A) THE CURVE OF TRAINING ERROR(B) THE X-Z VIEW OF FITTING RESULT(C) THE X-Z VIEW OF GROUND TRUTH(E) THE Y-Z VIEW OF FITTING RESULT(F) THE Y-Z VIEW OF GROUND TRUTH



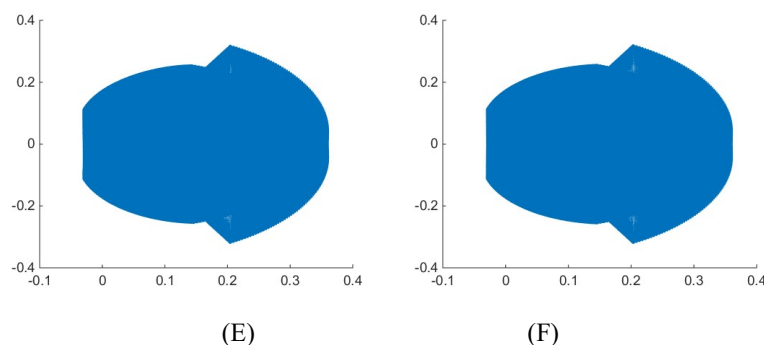


FIGURE IX. THE RESULTS OF RBF NEURAL NETWORKS TRAINED WITH 300 SAMPLES: (A) THE CURVE OF TRAINING ERROR(B) THE X-Z VIEW OF FITTING RESULT(C) THE X-Z VIEW OF GROUND TRUTH(E) THE Y-Z VIEW OF FITTING RESULT(F) THE Y-Z VIEW OF GROUND TRUTH

1) The RBF network is preferred when the training samples are few. Because training samples for BP network usually need to reach $1e4$, but RBF only need a few hundred samples.

2) The BP network is preferred when the computing resources are insufficient. RBF will increase the number of neurons dynamically during training. In order to achieve the fitting, the number of neurons in the end is basically the same as the amount of data, which needs huge computational cost.

ACKNOWLEDGMENT

The research is sponsored by the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (Intelligent Single Arm Robot Based on Visual Servo and Motor Imagery, No. Z2017030).

REFERENCES

- [1] M.R. Pedersen, L. Nalpantidis, R.S. Andersen, et al, "Robot skills for manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282-291, 2016.
- [2] P. Chen, L. Liu, F. Yu, et al, "A Geometrical Method for Inverse Kinematics of a Kind of Humanoid Manipulator," *Robot*, vol. 34, pp. 85-90, 2012.
- [3] J. Chen. Research on Real-Time Simulation System for Multiple DOF Robot Arm. Harbin Engineering University, 2012.
- [4] L. Long. Optimal Design Method of the manipulator structure based on the mobile platform. Wuhan University of Technology, 2012.
- [5] X.H. Li, Y.C. Guo, J. Zhang, et al, "Inverse Kinematics Solution and Verification of Modular 6-DOF Manipulator," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 44, pp. 246-251, 2013.
- [6] X.S. Li, H.B. Shi, R.C. Li. An improved analytical method of inverse kinematics of a humanoid robot. 2013 International Conference on Future Computer and Information Technology, Tianjin, WIT Transactions on Engineering Sciences, pp. 563-571, 2013.
- [7] H.B. Tian, H.W. Ma, J. Wei, "Workspace and Structural Parameters Analysis for Manipulator of Serial Robot," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 44, pp. 196-201, 2013.
- [8] K.S. Yi. Design and Experimental Research of Variable Stiffness Joint of Flexible Manipulator. Harbin Institute of Technology, 2016.
- [9] S.G. Yang. Multiple Degrees of Freedom Mechanical Arm Kinematics and Dynamics Research in Vehicle. Hunan University, 2016.
- [10] X.Y. Kang. The Accuracy Reliability Research and Simulation of Six DOF Serial Mechanical Arm. Guangdong University of Technology, 2016.

[11] V.Y. Kreinovich, "Arbitrary nonlinearity is sufficient to represent all functions by neural networks: a theorem," *Neural Networks*, vol. 4, pp. 381-383, 1991.

[12] J. Moody, C. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computing*, vol. 1, pp. 281-294, 1989.