

Received April 28, 2020, accepted May 12, 2020, date of publication May 18, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995172

# Low Delay Control Algorithm of Robot Arm for Minimally Invasive Medical Surgery

XIUMEI TIAN<sup>1</sup> AND YAN XU<sup>2</sup>

<sup>1</sup>Department of Automation, Heze University, Heze 274015, China

<sup>2</sup>Department of Electric Power, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

Corresponding author: Xiumei Tian (hzutxm@163.com)

**ABSTRACT** Minimally invasive surgical robots have received more and more attention from medical patients because of their higher surgical accuracy and higher safety than doctors. Minimally invasive surgery is rapidly revolutionizing the treatment of traditional surgery. In order to solve the problem that the surgical robot has a redundant degree of freedom, which makes the kinematics solution more complicated, this paper analyzes the kinematics of the coordinate system block. Aiming at the problem that the strategy search algorithm needs to re-learn when the target pose changes, a convolutional neural network control strategy is studied and constructed. By designing the structure of the convolutional neural network visual layer and motor control layer, the loss function and sampling of the training process are established. Aiming at the problem of long training time of convolutional neural network, an effective pre-training method is proposed to shorten the training time of the neural network. At the same time, the effectiveness of the above method and the end-to-end control of the convolutional neural network strategy are verified through simulation experiments. The physical structure of the manipulator body is analyzed, and the forward and inverse kinematic equations of the manipulator are established by the D-H method. Monte Carlo method was used to analyze the working space of the manipulator, and low-latency control and simulation experiments were carried out on the movement trajectory of the manipulator in joint space and Cartesian space. The results show that the low-latency control algorithm in this paper is effective to control the mechanical arm of the minimally invasive medical surgery robot.

**INDEX TERMS** Invasive medical surgical robot, kinematics analysis, convolutional neural network, low trajectory control.

## I. INTRODUCTION

Minimally invasive surgery originates from the development of traditional open surgery [1]. Traditional surgery has a large wound, the incision is generally greater than 100mm, and the scar is obvious [2]. It is easy to cause tissue damage at the patient's incision and trigger complications of tissue infection. Minimally invasive surgery changes many of the problems of the traditional surgery mentioned above. It uses small incisions on the body surface (generally less than 10mm) to penetrate surgical instruments and related medical equipment into the body, and performs the surgery under the guidance of imaging technology [3]–[5]. The wound of the patient is very small, and the recovery time can be greatly reduced. At present, minimally invasive surgery has been widely used

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei<sup>1</sup>.

in various fields of surgery, such as hepatobiliary surgery, cardiothoracic surgery, gynecology, and urology [6], [7].

The Endo Assist surgical robot system was developed by Armstrong Medical Center in the United Kingdom [8]. The surgical robot system can replace the assistant in the operation to hold the endoscope. The robot's controller has an inductive structure inside, making the controller a headgear-like form. During the operation, the attending doctor can put the controller on the head. The controller can detect the movement of the head and transfer the control information to the robot, so that the flexible control of the robot holding the endoscope can be easily achieved without occupying additional surgery operating space [9]–[11]. Computer Motion developed the ZEUS robot system in the late 1990s and introduced it to the market [12]. The robot system can be applied to minimally invasive surgery of the thorax and abdomen, such as the correction of

cardiac vascular malformations with a greater difficulty. The structure of the surgical robot adopts an integrated method, and according to the needs of the operation, the mechanical arm holding the mirror and the mechanical arm holding the surgical instrument are assembled in a suitable position on the operating table [13]–[15]. The key for the robot to complete the desired work according to people's wishes is the excellent control strategy [16]. As the earliest practical industrial controller, PID controller is still the most widely used industrial controller in the world because of its simple structure and easy adjustment of controller parameters [17], [18]. With the continuous development of control theory, traditional PID controllers often cannot meet the requirements of high precision and rapidity of the system [19]. Therefore, the current PID controllers derive many improved PID controllers by combining with other controllers, such as experts system tuning PID controller, neural network PID controller, multivariable decoupling PID controller and PID controller tuning based on various optimization algorithms, etc [20], [21]. These improved PID controllers combine the advantages of various controllers, and the control performance is significantly improved [22]. Many scholars at home and abroad have done a lot of research work on the PID controller in the trajectory tracking of the robotic arm [23]–[25]. Relevant scholars make full use of the simple structure of the PID controller and design the PID controller to realize the trajectory tracking control of the 3-DOF parallel manipulator [26], [27]. The PID parameters are set by Zeigler Nichols and genetic algorithm, and the tracking effect is significantly improved compared with the traditional PID. To solve the problem of high-precision trajectory tracking of double-link robots under model uncertainty and external disturbances, relevant scholars have designed a sliding mode control strategy that includes fuzzy tuning PID sliding mode surfaces [28]. This method combines fuzzy gain tuning, the robustness of the sliding mode controller and the rapid response characteristics of the PID, which effectively reduces the chattering caused by the sliding mode controller and improves the stability of the system.

In the trajectory tracking control of the robotic arm, fuzzy control usually assists other control strategies to achieve the control purpose. Relevant scholars have designed a fuzzy auto disturbance rejection controller for the trajectory tracking of robotic arms under the influence of uncertain factors such as unmodeled dynamics of the system [29]. The fuzzy controller is used to set the controller parameters and improve the robustness of the system. Relevant scholars have proposed a new fast iterative learning algorithm for manipulators with initial errors [30], [31]. The algorithm adds the deviation signal of two adjacent iteration errors at the same time to the conventional open-loop P-type iterative learning control, which effectively solves the problem of iterative initial state migration of the manipulator system, but the system convergence speed still needs to be improved. Related scholars have designed a discrete iterative learning control strategy with variable forgetting factors [32]. In this algorithm, the variable

forgetting factor is designed as the reciprocal form of the number of iterative learning times, and the system is analyzed for convergence based on the norm theory. Simulation results show that the control algorithm has significantly improved the convergence speed [33]. The doctor uses the master-slave control method when performing surgery with the "miao hand". The master-slave motion has an adjustable proportional mapping relationship, and the slave hand is equipped with a torque sensor to feedback the force signal to the master hand in real time. The doctor can feel the force of the robot through the main hand. In addition, researchers have designed several different surgical instruments for the system, each of which has different uses, such as clamps, sutures, and tissue processing.

In this paper, the D-H method is used to establish the reference coordinate system of each joint of the robot. Furthermore, according to the characteristics of surgical robots, by partitioning the coordinate system, the surgical robot in this paper has 6 positioning degrees of freedom and 1 redundant degree of freedom. This makes the kinematics solution problem more complicated and affects the response speed of the master-slave operation. Based on the idea of differential transformation, an equivalent differential transformation method suitable for the rapid forward and inverse kinematics of surgical robots in this paper is proposed, which successfully solves the problem that the traditional inverse transformation method has multiple solutions and transcendental functions that affect the response speed. Specifically, the technical contributions of this paper can be summarized as follows:

First: Aiming at the difficult design problem of the convolutional neural network strategy structure, we construct a reasonable structure of the visual layer and the motor control layer and design the visual layer structure to solve the feature drowning problem of the motor control layer, and carry out the training setting of the convolutional neural network.

Second: For the problem of long training time for convolutional neural networks, we design effective pre-training methods and formal training methods. The effectiveness of the method and the end-to-end control effect of the manipulator convolution neural network control strategy and the generalization ability of the target pose are verified through simulation experiments.

Third: For the establishment of the mathematical model of the multi-degree-of-freedom manipulator and the study of the low-delay control algorithm of the trajectory, we take the multi-degree-of-freedom manipulator as the control object and use the D-H method to establish the forward and inverse kinematics model of the manipulator. Based on this, the low-latency control of the trajectory of the manipulator in joint space and Cartesian space is analyzed.

The rest of this paper is organized as follows. Section 2 carries out the kinematic analysis of the minimally invasive surgical robot, and section 3 studies the control algorithm of the convolutional neural network of the manipulator.

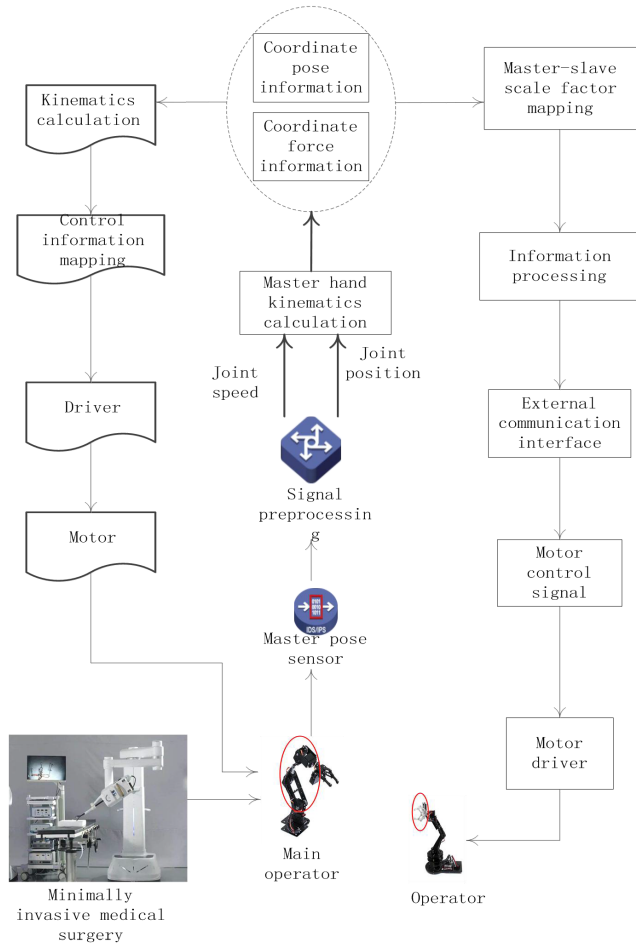


FIGURE 1. Schematic diagram of control system structure and function.

Section 4 carries out the simulation of low-latency trajectory control of the robot arm. Section 5 summarizes the full text and points out future research directions.

II. KINEMATIC ANALYSIS OF MINIMALLY INVASIVE SURGICAL ROBOT

A. CONFIGURATION OF MINIMALLY INVASIVE SURGICAL ROBOT

The minimally invasive surgical robot consists of three mechanical arms from the hand. The mechanical arm in the middle is used to clamp the endoscope, which is called the endoscope mechanical arm, and the other two mechanical arms are used to clamp the surgical instrument for the surgical operation, which is called the instrument mechanical arm. The structure of the three arms is basically the same. Since the mechanical arm of the instrument needs to be operated, the surgical instrument part at the end of the mechanical arm has three more degrees of freedom than the endoscope part at the end of the mechanical arm of the endoscope. The schematic diagram of the hardware structure and function of the control system is shown in Figure 1.

The joint configuration of the mechanical arm of the instrument is a tandem joint configuration, with a total of 6 joints.

Joint 1 is a linear motion joint, which realizes the overall lifting of the robot arm; Joint 2 is a rotary joint, without a driving motor inside, and is a passive joint used to adjust the arm shape before surgery to prevent the collision of the end of the robot arm during surgery. After the adjustment is completed, the joint is locked by the electromagnetic clutch on this joint; joint 3 is the slewing joint, joint 4 is the swing joint, and the internal parallelogram mechanism ensures the fixed-point movement of the telecentric point of the surgical instrument, and joint 7 is the linear motion joint. The linear movement of the terminal surgical instrument along the insertion point is realized. Joints 5, 6 cooperate so that the end of the surgical instrument can reach any position in the surgical area. The slewing joint 5 and the swing joint 6 have electromagnetic clutches inside, which can realize both active and passive control methods. Before the operation, a passive method is used to facilitate the doctor to adjust the posture of the surgical instrument when it is inserted; after the preoperative adjustment is completed and the operation is started, the two joints are converted into the active control mode and receive the control signal sent by the main hand.

The end of the mechanical arm of the instrument has three degrees of freedom to control the posture of the end, so as to achieve actions such as grasping, cutting, shearing, pinching and so on. The end of the endoscope mechanical arm has only one degree of freedom to rotate around its own axis.

B. POSITIVE KINEMATICS OF MINIMALLY INVASIVE SURGICAL ROBOTS

If the variable of each joint of the robot is known, the process of finding the position (p<sub>x</sub>, p<sub>y</sub>, p<sub>s</sub>) of the corresponding end effector wrist point in the base coordinate is called the kinematics positive solution.

According to the D-H notation, the homogeneous transformation matrix of the robot end effector relative to the base coordinates can be obtained as:

$${}^0T_7 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 {}^6A_7 \tag{1}$$

Considering joint 1, joint 2, joint 3, and joint 4, these four joints are fixed after the preoperative adjustment and the operation starts, the homogeneous transformation matrix of the coordinate system {4} relative to the base coordinate system {0} is a fixed value. After the surgical robot has been adjusted before operation and before the master-slave control, the movements of these four joints can be read in and calculated in advance through the auxiliary encoders in joint 1, joint 2, joint 3, and joint 4, which can reduce the master-slave control. The amount of real-time calculations is to speed up the master-slave response speed.

$${}^0T_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \tag{2}$$

The homogeneous transformation matrix of the robot end effector relative to the base coordinate is:

$${}^0T_7 = {}^0T_4 {}^4A_5 {}^5A_6 {}^6A_7 \tag{3}$$

In this paper, the purpose of solving the robot’s positive kinematics is to obtain the end effector wrist point coordinates  $\vec{p}_0$  according to the joint variable solution, the value is:

$$\vec{p}_0 = {}^0T_7 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (4)$$

**C. INVERSE KINEMATICS OF MINIMALLY INVASIVE SURGICAL ROBOTS**

If the position of the robot end effector wrist point in base coordinates ( $p_x, p_y, p_s$ ) is known, the process of finding the corresponding joint variable is called the inverse kinematics solution. Because the robot designed in this paper has a redundant degree of freedom, and its kinematics has an infinite number of inverse solutions, the rotation angle of the joint 4 needs to be determined according to other constraints, and the calculation is cumbersome. Considering joint 1, joint 2, joint 3, and joint 4, these four joints are fixed after the preoperative adjustment and the operation starts, the homogeneous transformation matrix of the coordinate system {4} relative to the base coordinate system {0} is a fixed value. You can first convert the position of the robot end effector wrist point in base coordinates ( $p_x, p_y, p_s$ ) to the position of the wrist point in the coordinate system {4} ( $p'_x, p'_y, p'_s$ ), and then in the coordinate system {4}, {5}, {6}, {7} inverse kinematics solution to obtain the corresponding joint variables. This method not only solves the tedious problem caused by the redundant degrees of freedom of the robot, but also reduces the amount of calculation and speeds up the response speed of the master and slave.

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_s \\ 1 \end{bmatrix} = ({}^0T_4)^{-1} \begin{bmatrix} p_x \\ p_y \\ p_s \\ 1 \end{bmatrix} \quad (5)$$

According to the position of the robot end effector wrist point in the coordinate system {4}, we get:

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_s \\ 1 \end{bmatrix} = {}^4A_5 {}^5A_6 {}^6A_7 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

The inverse solution obtained from the inverse transform method shows that there are four combinations of solutions, that is to say, to reach a certain posture from the end of the hand, the four situations solved can meet the requirements, but due to the limitations of the mechanical arm structure, some are even impossible. Because the joints of the robot arm cannot move within the entire 360° range. If there are multiple solutions, the shortest set of solutions is generally selected.

Because the inverse transform method has multiple solutions, it is necessary to judge the optimal solution in real time, and there is also an arc tangent transcendental function, which affects the response speed of the master-slave control, so it is

not suitable for the fast master-slave following requirements in this paper.

The Jacobian matrix has an important position in robot kinematics. It converts the speed of the robot’s single joint into the speed of the robot’s end effector, which can be regarded as the transmission ratio of the movement speed from the joint space to the operation space. The equation of motion of the operating arm is:

$$\dot{x} = \dot{x}(q) \quad (7)$$

It represents the displacement relationship between the operating space  $x$  and the joint space  $q$ . Taylor expansion of the above formula at a fixed point yields:

$$x(q + \Delta q) = x(q) + \frac{\dot{x}(q)}{1!} \Delta q + \dots + \frac{x^n(q)}{n!} (\Delta q)^n \quad (8)$$

Thus, the relationship between the position increment  $x$  of the end of the operating arm in Cartesian space and the angle increment  $q$  of the joint space is:

$$\Delta x = x(q + \Delta q) - x(q) = J(q) \Delta q + e \quad (9)$$

where  $e$  is the higher order error, ignoring the higher order error, we get:

$$\Delta x = x(q + \Delta q) - x(q) = J(q) \Delta q \quad (10)$$

The Jacobian matrix  $J(q)$  is a  $6 \times n$  matrix, where  $n$  is the number of degrees of freedom of the robot. The first three rows represent the linear velocity coefficients, and the last three rows represent the angular velocity coefficients. Therefore, the Jacobian matrix can be written as a block matrix:

$$J = \begin{bmatrix} J_L \\ J_A \end{bmatrix} = \begin{bmatrix} J_{L1} & J_{L2} & \dots & J_{Ln} \\ J_{A1} & J_{A2} & \dots & J_{An} \end{bmatrix} \quad (11)$$

In the formula,  $J_{Li}$  and  $J_{Ai}$  represent the three-dimensional linear velocity and angular velocity coefficients caused by the  $i$ th joint variable, respectively. The elements in row  $i$  and column  $j$  of  $J(q)$  are:

$$J_{ij}(q) = \frac{\partial x_i(q)}{\partial q} \quad (12)$$

It can be seen from the above equation that for a given  $q$ , the Jacobian matrix  $J(q)$  is a linear transformation from the joint space velocity  $q$  to the operation space velocity  $x$ .

Inverse kinematics can be simplified to only require solving three degrees of freedom inverse problems, and only need to analyze the three-dimensional linear velocity problem at the end of the robot caused by the  $i$ -th joint variable, so the Jacobian matrix  $J(q)$  is reduced to a  $3 \times 3$  matrix.

The joint variable expression obtained by the inverse kinematics transformation method contains arc tangent and other transcendental functions, and contains multiple solutions. It is necessary to judge and select the optimal solution in real time, which requires a lot of calculations, resulting in a reduction in the real-time control. The equivalent differential transformation method replaces the instantaneous velocity of



the joint with the displacement in a small time period, which can reduce the number of inverse kinematics of the robot and increase the response speed of the master and slave.

### III. RESEARCH ON THE CONTROL ALGORITHM OF THE MANIPULATOR CONVOLUTIONAL NEURAL NETWORK

#### A. THE ESTABLISHMENT OF THE CONTROL STRATEGY OF THE MANIPULATOR CONVOLUTIONAL NEURAL NETWORK

When constructing the control strategy of the convolutional neural network of the robotic arm, the input of the convolutional neural network needs to be determined first. For the robotic arm controller optimized by the strategy search algorithm, the input is the joint information (joint angle, angular velocity) and end effector information (end effector position, speed) of the robot arm at the current moment, and the output is the joint of the robot arm at the current moment control amount. Since the training of the convolutional neural network strategy requires the controller obtained by the strategy search algorithm to provide supervision, its input must also contain all the information contained in the controller.

Since the convolutional neural network has very powerful image processing capabilities, by acquiring images of the end effector of the robot arm and the target position at each control moment and using it as the input of the convolution layer, the neural network can search the strategy during the training process. Under the supervision of the algorithm, the network weights and offsets are adjusted to obtain the relative pose information and some more advanced features of the end effector and target in the input image; compared to the end effector information, the joint information of the robot arm cannot pass through the camera image intuitively. Therefore, when building a convolutional neural network strategy, the joint angle and angular velocity information of the robot arm are directly used as the input of the fully connected layer of the neural network. Through the above analysis, the convolutional neural network strategy constructed in this paper and its control idea for the robot arm are shown in Figure 2.

The constructed convolutional neural network strategy includes two parts: the visual layer and the motor control layer. The visual layer is mainly composed of the convolution layer and the pooling, and the motor control layer is composed of the fully connected layer. At each control moment of the robotic arm, the visual layer of the convolutional neural network strategy first processes the camera image at that moment and outputs feature information vectors, and then the neural network combines the feature information vectors output by the visual layer and the robotic arm at that moment. The joint information vectors are combined and used as input to the motor control layer. Finally, the joint control amount of the robot arm at the current moment is obtained through the motor control layer.

After obtaining the joint control amount of the robot arm at the current time through the convolutional neural network strategy, the control amount is sent to the robot arm control box by the host computer and the robot arm is smoothly

moved to the corresponding joint position through joint interpolation and the robot arm is completed at the same time. In the item placement task, the convolutional neural network strategy continuously performs the above control process, and finally causes the robot arm to complete the placement of the corresponding item.

By constructing the above convolutional neural network strategy, not only can the robotic arm have a certain adaptability to the placement of the paper, but also this neural network strategy does not require the camera to be calibrated or any processing on the original image. This end-to-end mechanism that directly uses the original data and obtains the final output is a major feature of deep reinforcement learning algorithms in robot control, and it is also the most prominent advantage of the convolutional neural network strategy in this paper when controlling the robotic arm.

#### B. STRUCTURAL DESIGN OF CONVOLUTIONAL NEURAL NETWORK

In the construction of the visual layer of the convolutional neural network, the convolution-pooling-convolution network structure is used here to process the input image and a fully connected layer is added at the end of the visual layer to combine image features and reduce the dimension of the image features, and at the same time scale the value of the image features to make it equal to the value of the joint state of the robot arm; then the convolutional neural network will the image features extracted by the visual layer and the joint of the robot arm. The state information is combined and used as the input of the motor control layer. In the construction of the motor control layer, two fully connected layers are used to fit the distribution of the strategy search controller; after the motor control layer. The joint control amount of the robot arm at this moment is output.

After the camera image is processed by visual layer convolution and pooling, if the obtained feature map is directly combined with the joint state information of the robot arm and used as the input of the motor control layer, it will make the convolutional neural network strategy difficult in training convergence. The reason for this phenomenon is that after the convolution and pooling process of the camera image, the excessive number of features will make the joint state information of the robot arm be overwhelmed by the extracted image features. At the same time, the input dimension of the motor control layer is too large. It is difficult to obtain the mapping relationship between the input features and the control amount of the robot arm joint during training.

Therefore, when designing the visual layer structure of the convolutional neural network, a fully connected layer is added at the end of the visual layer, which mainly has the following two functions. The addition of fully connected layer 1 can greatly reduce the input dimension of the motor control layer and prevent the joint state information of the robot arm from being overwhelmed by image features. Under the strategy structure of the convolutional neural network, the feature map output by the convolutional layer 2

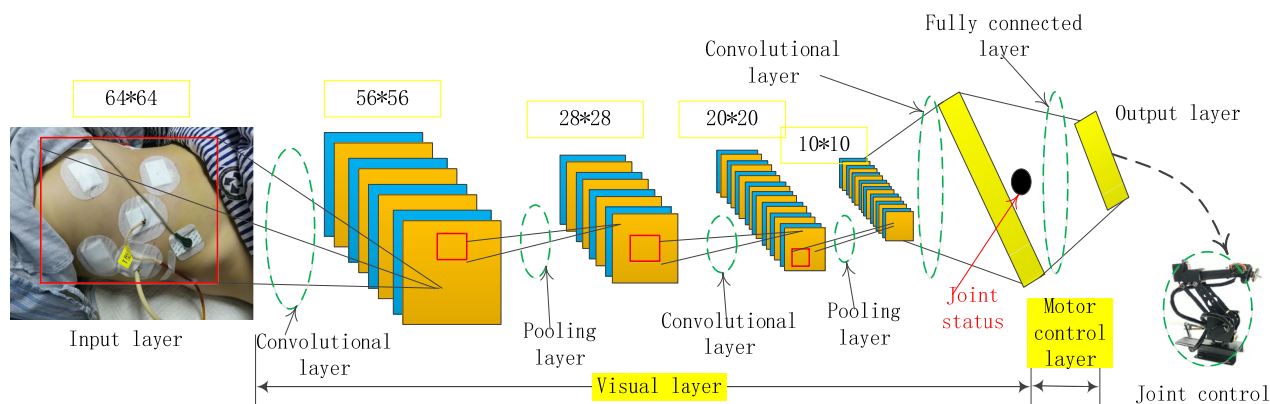


FIGURE 2. Manipulator convolution neural network control strategy.

after the fully connected layer 1 has a feature dimension reduced to 28, which is more quantitatively matched with the 12-dimensional joint state information of the UR10 robot arm. Secondly, the addition of fully connected layer 1 can make the extracted image features numerically comparable to the joint state information of the robotic arm, and enable the convolutional neural network to better perform parameters on the motor control layer when performing random gradient descent optimization.

**C. CONVOLUTIONAL NEURAL NETWORK TRAINING SETTINGS**

In the convolutional neural network, the loss function not only evaluates the quantitative index of the output of the neural network, but its definition will directly determine the optimization direction of the weights and biases of each layer during the training of the convolutional neural network, and ultimately determine the convolutional neural network.

For the convolutional neural network constructed in this paper, the training process is carried out under the supervision of the strategy search algorithm. Therefore, in order to enable the robotic arm to complete the prescribed placement task at the training position under the convolutional neural network strategy control, the positions that have not appeared in the training are also adaptable. The loss function needs to be designed so that the convolutional neural network strategy can imitate the robotic arm controller that provides supervision. At the same time, the parameters of the visual layer and the motor control layer are to obtain a certain generalization ability.

In the strategy search algorithm, for each task, the algorithm will only run the controller optimized for the last cycle and sample N times when sampling. When training the convolutional neural network under the strategy search algorithm, although the original sampling method can also be used to optimize and obtain the convolutional neural network strategy corresponding to the task, but the convolutional neural network can also control the robotic arm for sampling, so the original sampling method caused a waste of convolutional neural network resources. Based on the above

reasons, the algorithm uses the greedy sampling method to fuse the convolutional neural network and the controller when training the convolutional neural network strategy, so that the algorithm can fully utilize the convolutional neural network strategy and make the neural network get better optimization effect.

In each learning environment, the robotic arm first samples each under the control of the current controller and convolutional neural network strategy, because the algorithm needs to compare the current controller and convolutional neural network strategy by calculating the cost of the above two trajectories, so in the two samplings, no noise can be added to the output actions of the controller and the convolutional neural network strategy. After calculating the cost of the two trajectories obtained, the corresponding sampling method is used as the main sampling of greedy; after removing the more expensive trajectories from this sampling, the algorithm will finally run a new greedy sampling method to control the robotic arm and sample N2 times. Action noise is added to better achieve the purpose of exploring the manipulator state action space near the current trajectory.

In the above sampling method, N is the number of samplings required for each task, and its value is very important. If N is too large, the efficiency of the algorithm will decrease and the learning cost will increase; if N is too small, the algorithm will not converge due to insufficient exploration of the manipulator’s state action space. In this paper, the Gaussian mixture model is used as the prior of the local environment dynamic model in the strategy search algorithm, which effectively reduces the number of samples required. Therefore, the actual program passes the test and the final value of N is 5.

In the constructed  $\epsilon$ -greedy sampling method, the values of hyperparameters also need to be set.  $\epsilon$ -greedy sampling method is one of the more commonly used sampling methods for reinforcement learning. It uses the optimal strategy under the current cycle of the algorithm as the main sampling and randomly selects all sampling methods with a certain probability at each sampling time to make full use of existing knowledge and better explore the purpose of state

action space. In the  $\epsilon$ -greedy sampling method,  $\epsilon$  is the hyperparameter of the sampling method, which indicates the approximate probability of selecting the main sampling method as the judgment criterion for the output action of the execution object. The larger  $\epsilon$  is, the lower the efficiency of exploring the state action space of the execution object at the same time. In this algorithm, because the action noise is added when the robotic arm is controlled by the  $\epsilon$ -greedy method for sampling, the purpose of exploring near the current optimal trajectory is achieved, so the setting of the hyperparameter value does not need to be too low. The final value in is 0.95. Since there are only two sampling methods in this algorithm, at each sampling moment of the robot arm, the probability of the algorithm using the current  $\epsilon$ -greedy's main sampling method is 0.964, and the probability of using the auxiliary sampling method is 0.036.

**D. CONVOLUTIONAL NEURAL NETWORK FORMAL TRAINING**

In the convolutional neural network strategy, the role of the visual layer is to obtain the relative pose information of the end effector of the robot arm and the target position and some other advanced features by processing the camera image. Therefore, when pre-training the visual layer of the convolutional neural network, you can add a fully connected layer after the neural network visual layer to adjust the output dimension of the network and make the pre-training model output the absolute position of the target position in the task. This enables the convolutional neural network to first learn the absolute position information of the target and the initial information of the end effector before formal training, so that the neural network vision layer and motor control layer parameters can be quickly optimized during formal training.

After training the pre-trained model of the convolutional neural network by using the image data of the training set and the method of random gradient descent, the algorithm will use the test set to test the current neural network model. The parameters of the visual layer part of the neural network pre-training model are saved, and the pre-training process is completed at the same time.

The purpose of this paper to build a convolutional neural network strategy is to enable the robotic arm to complete end-to-end control under the trained convolutional neural network strategy and have certain adaptability to the placement of the paper. When conducting formal training, it is necessary to effectively use the supervision of the strategy search algorithm, so that the trained convolutional neural network has good generalization ability. The convolutional neural network training process is shown in Figure 3.

By fitting a linear multivariate Gaussian distribution around the current trajectory as the environmental dynamic model, it is already a high approximation of the environmental dynamic model of the robotic arm. However, due to the large dimension of the manipulator's state action vectors, the number of samples needed to fit the mean term coefficients and covariance matrix will be very large, resulting in

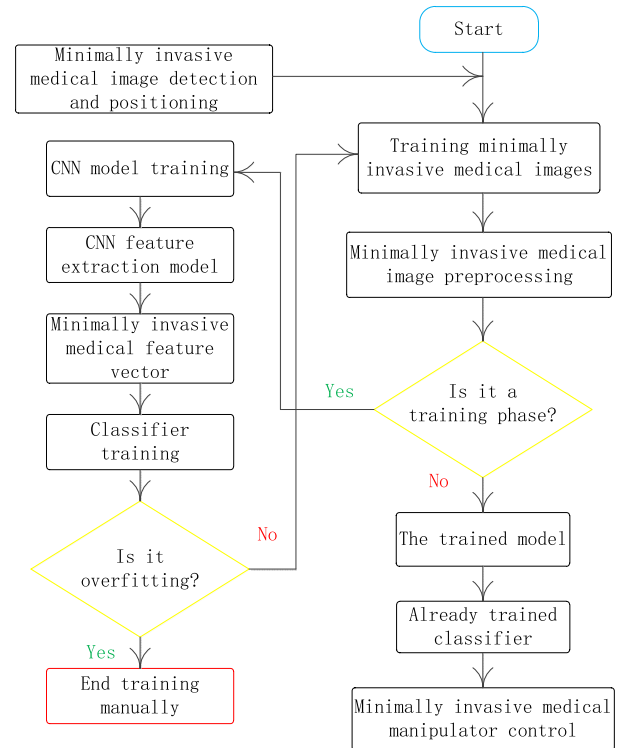


FIGURE 3. Convolutional neural network training process.

a decrease in the efficiency of the strategy search algorithm and an increase in learning costs. Therefore, in order to reduce the actual number of samples of the robotic arm, when fitting the local environment dynamic model, the algorithm first clusters the new and old samples by fitting the Gaussian mixture model, and uses the normal inverse Wicht prior and the maximum. A posteriori estimation reconstructs the mean and covariance of the newly acquired samples at each moment to reduce the number of samples actually taken by the robotic arm.

When the algorithm is training and testing the convolutional neural network strategy, the target position of the object placement task in simulation and actual experiments is limited to a square area, where the side length of the square area is about 38 cm in the actual experiment. The square area selected in the actual experiment is not very large. The main reasons are: first, the improvement of the generalization ability of the convolutional neural network needs to improve the network depth and the location and number of samples. Both methods will greatly increase the algorithm for the volume. In addition, when the home service robot performs the item placement task, it is adaptable to the target position of the above area and can solve many problems.

In order to make the convolutional neural network strategy adapt to any target position in the square area after training, the algorithm when training the convolutional neural network, selects the four positions in the square area and uses the strategy search algorithm to obtain the corresponding

position under the controller. Then, the obtained controller is used as the method of supervising the convolutional neural network strategy and optimizing the parameters of the neural network, and finally the generalization ability of the convolutional neural network strategy can meet the requirements after training. In the selection of training positions, the selected training positions are 1, 3, 7, and 9, and in actual experiments, the effects of optimizing the convolutional neural network due to the selection of 2, 4, 6, and 8 position controllers are more effective. So the training position was adjusted during the actual experiment. Since the optimized controller of the strategy search algorithm is a multivariate Gaussian distribution with a mean of linear form, the convolutional neural network will obtain the entire square area after optimizing the convolutional neural network strategy by obtaining the controller at the above position as a supervision. The distributed form of the controller finally makes the neural network adaptable to the position of the entire target area.

During the formal training of the convolutional neural network strategy, the parameters of the motor control layer are not pre-trained, which will cause the convolutional neural network to have too large a gradient of the parameters at the initial stage of training, resulting in the weights and offsets of each layer of the neural network. All of them are updated significantly, and finally the neural network visual layer parameters discard the feature information mined during pre-training and during the update process. Therefore, after using the pre-trained visual layer parameters and initialization methods to initialize the convolutional neural network visual layer and motor control layer parameters, the motor control layer parameters of the neural network are first trained. When training the parameters of the motor control layer of the convolutional neural network, the algorithm first fixes the parameters of the visual layer of the neural network, and then uses the samples obtained at the training position.

After initializing the convolutional neural network vision layer parameters through pre-training and first optimizing the neural network motor control layer parameters during formal training, the algorithm can formally train the convolutional neural network strategy end-to-end. During formal training, first of all, it is necessary to randomly sort all the samples obtained by the deep reinforcement learning algorithm at the training position to reduce the correlation of the adjacent data in the training set. Afterwards, by using the Adam stochastic gradient descent method to optimize the parameters of the convolutional neural network, each time the random gradient descent is performed, the algorithm will sequentially select 20 rearranged training set data to train the convolutional neural network strategy, and deep reinforcement learning algorithm performs 600 random gradient descents in the inner loop to update the parameters of the convolutional neural network. By using the above pre-training and formal training methods, the actual training time of the convolutional neural network strategy is about 30 minutes.

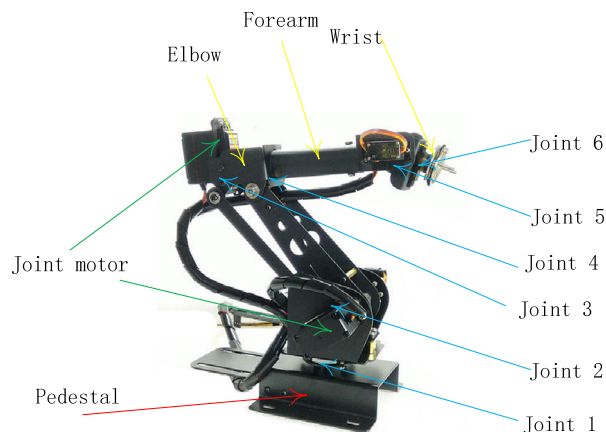


FIGURE 4. Robotic arm of minimally invasive medical surgery robot.

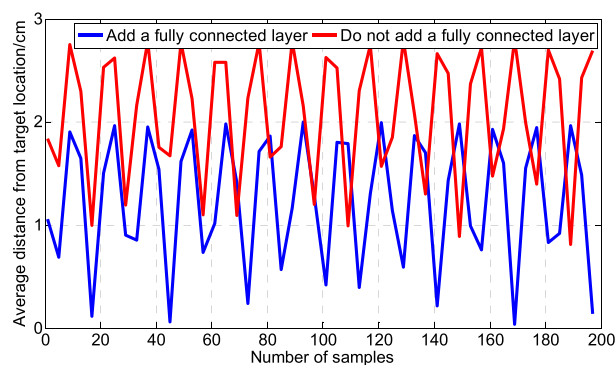


FIGURE 5. The learning curve before and after the convolutional neural network vision layer is added to the fully connected layer.

## IV. ROBOT ARM LOW-LATENCY TRAJECTORY CONTROL SIMULATION

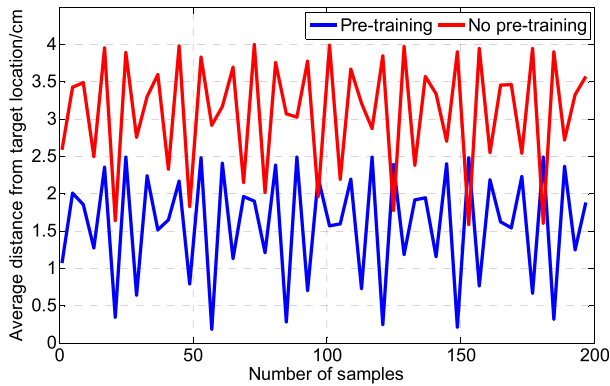
### A. CONVOLUTIONAL NEURAL NETWORK STRATEGY SIMULATION

When acquiring the input image in this simulation, after defining the camera pose using the official API, the size of the obtained camera image can be arbitrarily defined within the maximum image size range. Through adjustment and testing, the size of the original image was defined as  $900 \times 640$  during simulation, and the image was compressed to  $90 \times 64$  as the input of the convolutional neural network. The mechanical arm of the minimally invasive medical surgery robot is shown in Figure 4.

In this simulation, in order to enable the convolutional neural network strategy to be trained, it can control the robot end-to-end and adapt to the position of the hole in the specified area. The training position was also sampled 8 times. In the simulation, 8 simulation experiments were conducted for the two convolutional neural network strategies, and the learning curves of the 8 simulations were averaged to obtain the total learning curve, as shown in Figure 5.

As can be seen from Figure 5, when the fully connected layer is not added to the visual layer of the convolutional neural network strategy, after 200 samples of the deep reinforcement learning algorithm, the robot performs its axis entry under the control of the final convolutional neural



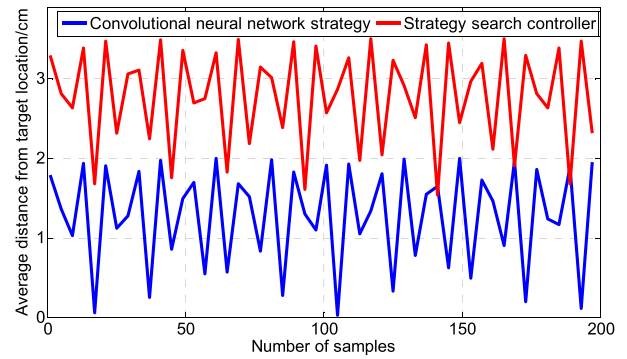


**FIGURE 6.** Convolutional neural network strategy learning curve before and after pre-training.

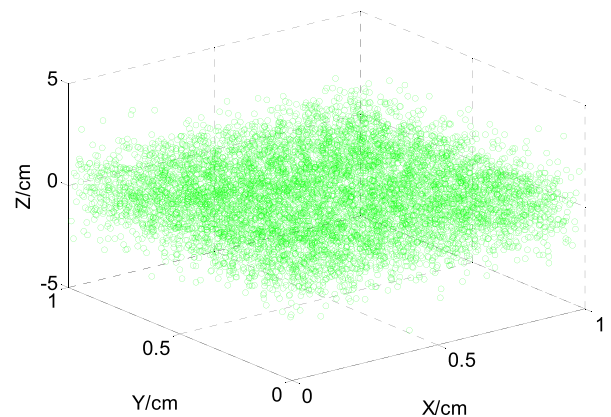
network strategy. The average error of the hole task still exceeds 2.5cm, and the specified placement task cannot be completed. When the convolutional neural network does not add a fully connected layer, the standard deviation of the average distance from the target position is also large, and the algorithm has a great randomness in the optimization effect of the convolutional neural network. After the fully connected layer is added to the visual layer, the average error of the robot performing the placement task can be lower than the case where the fully connected layer is not added, and the average distance from the target position with each iteration of the deep reinforcement learning algorithm. The difference will also be lower.

It can be seen from Figure 6 that during the first cycle of the deep reinforcement learning algorithm, the average target position error of the robot arm under the control of the pre-trained convolutional neural network strategy is significantly lower than that of the Under the control of the pre-trained convolutional neural network strategy, it even appears to be far away from the target position. This is because by pre-training the visual layer of the convolutional neural network, the neural network has obtained a large amount of information about the target position and the end effector of the robot arm during formal training, and the convolutional neural network without pre-training is the first in the algorithm. The above information cannot be extracted in the second cycle, which leads to a great difference in the learning effect of the two in the first cycle of the algorithm. In the subsequent cycle of the algorithm, when the robot arm executes the shaft-in hole task under the control of the pre-trained convolutional neural network strategy, its average target position error is also smaller than that of the neural network without pre-training.

After averaging the obtained simulation experimental data, the overall learning curve of the supervised strategy search controller and convolutional neural network strategy is shown in Figure 7. As can be seen from Figure 7, under the supervision of the strategy search algorithm and the convolutional neural network strategy in the deep reinforcement learning algorithm cycle, its control effect on the robotic arm is



**FIGURE 7.** Learning curve of controller and convolutional neural network strategy.



**FIGURE 8.** The reachable space of the multi-degree-of-freedom manipulator.

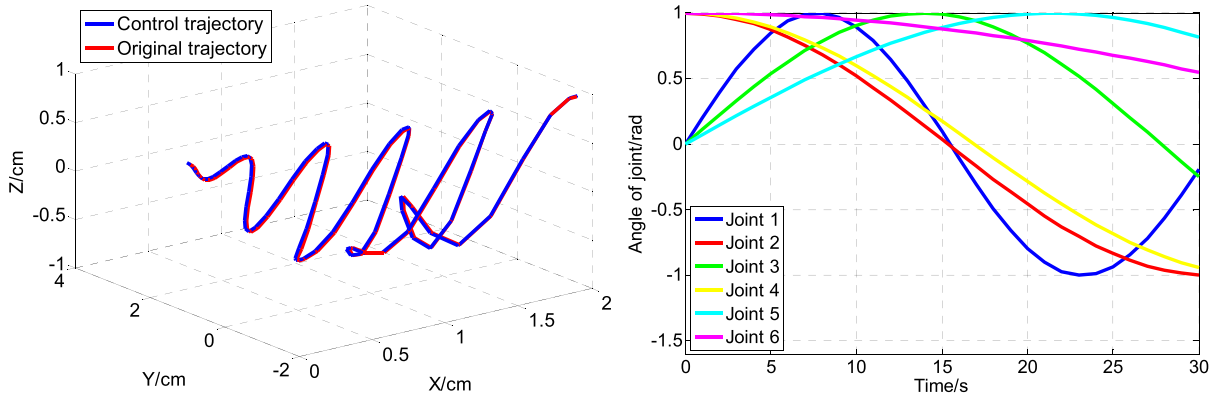
better than the controller that provides supervision, this is because the parameters of the current convolutional neural network are fully optimized after the defined stochastic gradient descent cycle.

## B. REACHABLE SPACE ANALYSIS OF MULTI-DEGREE-OF-FREEDOM MANIPULATOR

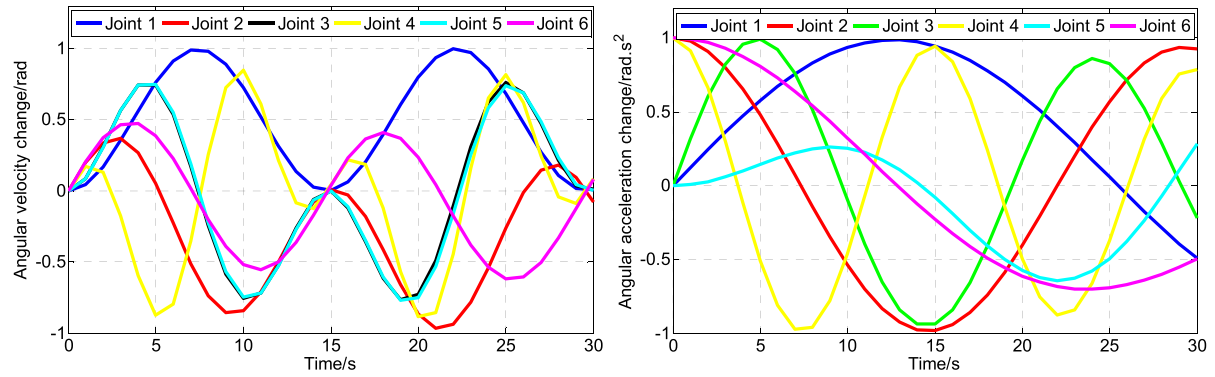
The reachable space of the multi-degree-of-freedom manipulator refers to the range that can be reached by the end position of the manipulator after the manipulator is fixed, and is an important indicator to measure the working ability of the manipulator. Generally speaking, this indicator should be carefully considered at the initial stage of robotic arm design and development.

There are many ways to solve the reachable space of the robotic arm. The most important ones are graphic method, mathematical analysis method and numerical method. In this paper, the Monte Carlo method is used to analyze the reachable space of the manipulator. Monte Carlo method is a typical numerical analysis method. Within the range of motion of the robot arm joint variables, a random function is used to generate a certain number of uniformly distributed random numbers. We substitute this random number (ie joint variable) into the robot arm end position vector, so as to obtain the space range that the robot arm end position can reach, and





(a) Low-latency control of the end trajectory of the robot arm during cubic polynomial planning (b) Joint angle change curve during cubic planning



(c) Joint angular velocity change curve during the third planning (d) Joint angular acceleration change curve during the third planning

FIGURE 9. Cubic polynomial low-latency trajectory control simulation results.

this range is the reachable space of the robot arm.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_1 s_{23} + c_1 (c_2 + 2) \\ s_1 c_{23} + s_1 (c_3 + 2) \\ c_{23} + s_2 + s_{23} \end{bmatrix} \quad (13)$$

The above formula means that the end position of the robot arm is in the space position. Where  $c_i$  represents  $\cos \theta_i$  and  $s_i$  represents  $\sin \theta_i$ . We use Matlab to generate 14000 random points to get its reachable space, as shown in Figure 8.

**C. CUBIC POLYNOMIAL LOW-LATENCY TRAJECTORY CONTROL ON JOINT SPACE**

Suppose the starting position of the movement of a joint of the robot arm is  $\theta_0$ , and the final position of the movement is  $\theta_f$ , and the speed of the joint is 0 at the start time. Then these four constraints can uniquely determine all parameters of a cubic polynomial. Let the cubic polynomial be:

$$\theta(t) = a + bt + ct^2 + dt^3 \quad (14)$$

The constraints are:

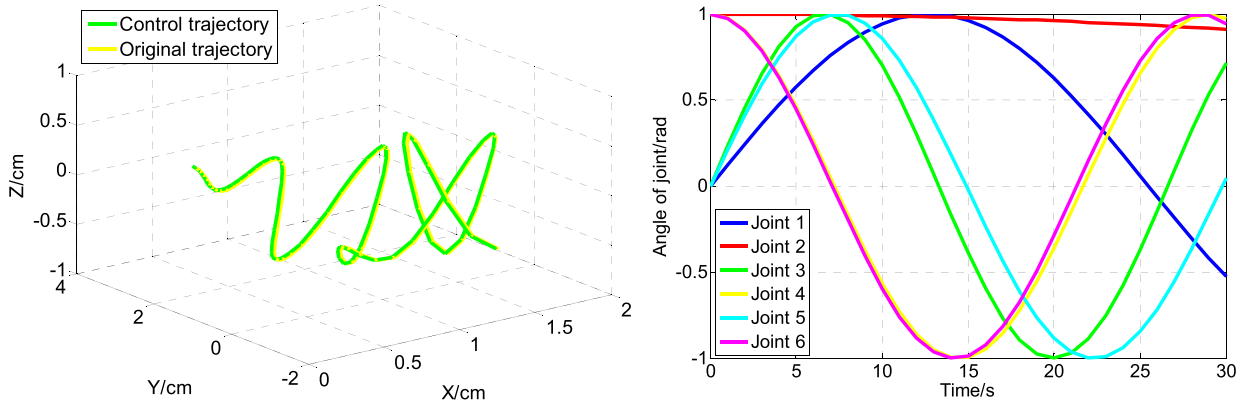
$$\begin{cases} \theta(t_0) = \theta_0 \\ \theta(t_f) = \theta_f \\ \dot{\theta}(t_0) = 1 \\ \dot{\theta}(t_f) = 1 \end{cases} \quad (15)$$

The following is a simulation experiment of the three-path low-latency control in the joint space. Suppose the simulation

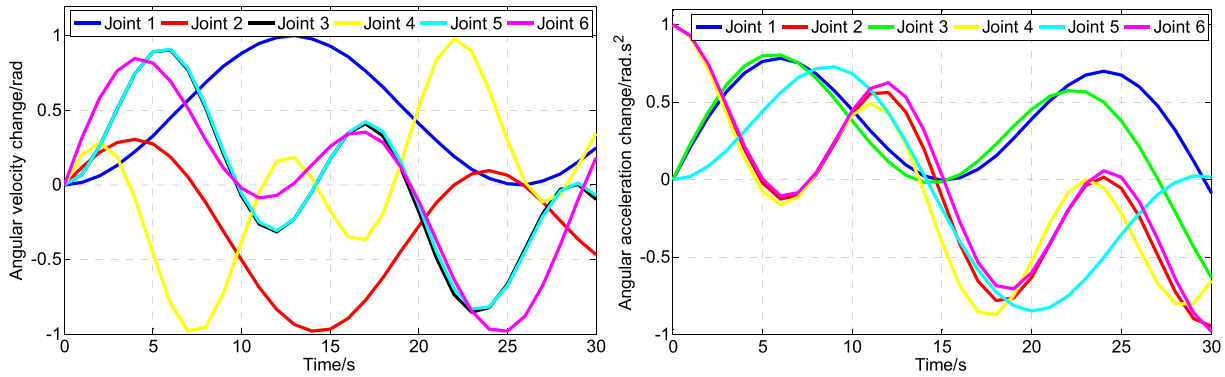
time is 30s, the starting point of the low-latency control of the trajectory is A. The joint variable matrix is  $[-0.5 \ -0.6 \ -0.7 \ -0.5 \ -0.7 \ 0]$ , and the intermediate point B is reached after 15 seconds. The end point C is reached after 30 seconds, and the joint variable matrix of the end point C is  $[0.5 \ -0.1 \ 0.4 \ 0.7 \ 0 \ 0]$ . Using the cubic polynomial programming method to calculate, we calculate the joint variable of each intermediate point during the movement of the robot arm and draw the position and posture of the point, and obtain a low-latency control simulation diagram of the motion trajectory of the end of the robot arm, as shown in Figure 9 (a). At this time, the curves of angle, angular velocity and angular acceleration of each joint are shown in Figure 9(b), 9(c) and 9(d).

**D. LOW-LATENCY TRAJECTORY CONTROL OF FIFTH-DEGREE POLYNOMIALS IN JOINT SPACE**

The three-item polynomial trajectory low-latency control is only a plan for joint angle and angular velocity, and there is no angular acceleration constraint, so the continuity of angular acceleration cannot be guaranteed. The discontinuous changes in angular acceleration will adversely affect the movement process of the robotic arm, and the joint motor will also be subjected to uncertain impacts, affecting the life of the motor. The fifth-degree polynomial is precisely corrected, and the constraint on the angular acceleration of the joint is added on the basis of the third-degree polynomial.



(a) Low-latency control of the trajectory of the end position of the robot arm during fifth degree polynomial planning (b) Variation curve of joint radian during fifth degree planning



(c) Joint angular velocity curve during fifth planning (d) Joint angular acceleration curve during fifth planning

FIGURE 10. Low-latency trajectory control simulation results of the fifth degree polynomial.

Although this method increases the amount of calculation, it ensures the stability of the joint motion process.

Suppose the start time of a joint movement of the robot arm is  $t_0$ , the end time of the movement is  $t_f$ , the start speed is  $\theta_0$ , and the end speed is  $\theta_f$ . Simultaneously we set the fifth degree polynomial as:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (16)$$

The coefficients of the fifth degree polynomial are:

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = 0.3 \cdot \ddot{\theta}_0 \\ a_3 = 10\theta_f - 10\theta_0 - 4\dot{\theta}_f t_f - 2\ddot{\theta}_0 t_f^3 \\ a_4 = 14\theta_f - 15\theta_0 - 3\dot{\theta}_f t_f - 4\ddot{\theta}_0 t_f^4 \\ a_5 = 6\theta_f - 5\theta_0 - 2\dot{\theta}_f t_f - \ddot{\theta}_0 t_f^5 \end{cases} \quad (17)$$

The following is a simulation experiment of the five-path low-latency control in the joint space. We set the simulation time to 15s, the starting point of the low-latency control of the track is A (the joint variable matrix of point A is  $[-0.7 \ -0.4 \ -0.3 \ 0.7 \ 0.2 \ 0]$ ), and the robot arm reaches the intermediate point B (the matrix is  $[-0.3 \ 0.2 \ -0.5 \ 0.4 \ 0.5 \ 0]$ ), it reaches the intermediate point C after 10 seconds (the joint variable matrix of C is  $[0.3 \ 0.6 \ -0.7 \ 0 \ 0.2 \ 0]$ ), and the end point D (the joint variable of D after 15 seconds) matrix is  $[0.7 \ 0.5 \ -0.3$

$0.2 \ 0.2 \ 0]$ ). Using the fifth degree polynomial programming method to calculate the joint variable of each intermediate point during the movement of the robot arm and draw the position and posture of the point, we obtain a low-latency control simulation diagram of the movement trajectory of the end of the robot arm, as shown in Figure 10 (a). At this time, the curves of angle, angular velocity and angular acceleration of each joint are shown in Figure 10 (b), 10 (c) and 10 (d).

From the simulation results of the above two plans, it can be found that the low-latency control of the trajectory in the joint space can ensure that the joint angle changes are continuous, but it cannot guarantee that the end of the robot arm moves along a specific trajectory in Cartesian space (right-angle space), which can not directly constrain the attitude of each point.

**E. LOW-LATENCY LINEAR TRAJECTORY CONTROL ON CARTESIAN SPACE**

Suppose the starting point coordinate of the motion line is  $p_0 (x_0, y_0, z_0)$ , the end point coordinate is  $p_1 (x_1, y_1, z_1)$ , and the movement speed of the end position of the robot arm is  $V$ , then the length of the straight line is:

$$L = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \quad (18)$$

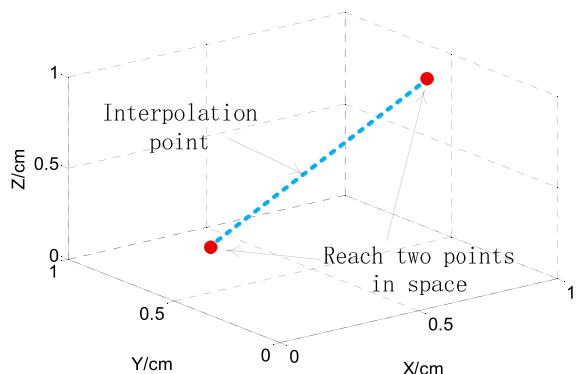


FIGURE 11. Interpolation curve of low-latency control of linear trajectory.

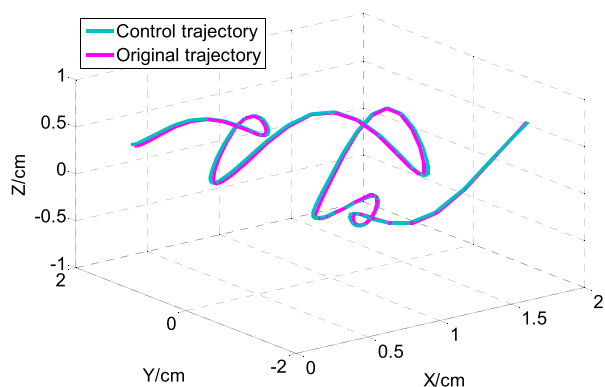


FIGURE 12. Low-latency control of the movement trajectory of the end of the robot arm in straight line planning.

Then the exercise time  $T$  is  $L / V$ . We define a total of  $N$  points to be sampled on a straight line of motion, define the  $f(t)$  value range as  $[0 \sim 1]$ , and it is an increasing function of  $f(0) = 0, f(1) = 1$ :

$$\theta = f(t) = f(i/N) \tag{19}$$

Then the linear trajectory interpolation point on Cartesian space can be expressed as:

$$\begin{cases} x_i = x_0 + (x_1 - x_0)\theta \\ y_i = y_0 + (y_1 - y_0)\theta \\ z_i = z_0 + (z_1 - z_0)\theta \end{cases} \tag{20}$$

The following is a simulation experiment of the low-latency control of the linear trajectory. The straight line planning of this subject is to stipulate the waist rotation joint, wrist pitch joint and wrist rotation joint to maintain a stable posture, that is, only the vertical arm joint and the horizontal arm joint are planned.

We take two points in the reachable space, set the planning time to 20S, and interpolate a total of 22 points between them. The interpolation curve at this time is shown in Figure 11.

The obtained joint variables are planned using the cubic polynomial trajectory trajectory method in the joint space, and the low-latency control simulation diagram of the end trajectory of the robot arm is shown in Figure 12.

## V. CONCLUSION

In order to solve the problem that the surgical robot in this paper has multiple degrees of freedom in positioning, which makes the kinematics solving problem more complicated and affects the response speed of the master-slave operation, this paper analyzes the kinematics by dividing the coordinate system. The structure of the convolutional neural network is designed. Through the construction of the visual layer and the motor control layer, the convolutional neural network completes the control of the mechanical arm and the neural network is improved during training by adding a fully connected layer to the visual layer. The initial settings and hyperparameters of the neural network are optimized. At the same time, for the problem of long training time of the convolutional neural network, the neural network training time is shortened by designing an effective neural network visual layer pre-training method. Simulation experiments verify the effectiveness of the convolutional neural network strategy's visual layer structure, sampling method, and pre-training method, as well as the end-to-end control effect of the neural network and its generalization ability for the target pose. The constructed convolutional neural network strategy can control the robot arm end-to-end and has a good generalization ability for the placement and posture of items. The Monte Carlo method is used to obtain the motion range of the multi-degree-of-freedom manipulator in three-dimensional space, that is, the working space of the manipulator. The trajectory low-delay control algorithm of cubic polynomial and quintic polynomial in joint space, and the low-delay control algorithm of straight line and arc trajectory on Cartesian space are simulated, and the simulation results are given respectively. In the future, the scope and flexibility of the surgical work of the telecentric mechanism that pulls the actuator and holds the endoscope will be comprehensively considered, and the freedom of the robot arm configuration will be further studied in order to expect the premise of meeting the requirements to reduce the freedom and mass of the robotic arm.

## REFERENCES

- [1] X. Wu, J. Wang, X. Sun, and W. Han, "Guidance for the treatment of femoral neck fracture with precise minimally invasive internal fixation based on the orthopaedic surgery robot positioning system," *Orthopaedic Surgery*, vol. 11, no. 3, pp. 335–340, Jun. 2019.
- [2] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1447–1453, Apr. 2019.
- [3] R. Trochimczuk, A. Łukaszewicz, T. Mikołajczyk, F. Aggogeri, and A. Borboni, "Finite element method stiffness analysis of a novel tele-manipulator for minimally invasive surgery," *Simulation*, vol. 95, no. 11, pp. 1015–1025, Nov. 2019.
- [4] X. Ma, C. Song, P. W. Chiu, and Z. Li, "Autonomous flexible endoscope for minimally invasive surgery with enhanced safety," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2607–2613, Jul. 2019.
- [5] A. K. Ahmed, C. C. Zygourakis, S. Kalb, A. M. Zhu, C. A. Molina, B. Jiang, A. M. Blitz, A. Bydon, N. R. Crawford, and N. Theodore, "First spine surgery utilizing real-time image-guided robotic assistance," *Comput. Assist. Surgery*, vol. 24, no. 1, pp. 13–17, Jan. 2019.
- [6] J.-H. Hsiao, J.-Y. Chang, and C.-M. Cheng, "Soft medical robotics: Clinical and biomedical applications, challenges, and future directions," *Adv. Robot.*, vol. 33, no. 21, pp. 1099–1111, Nov. 2019.

- [7] N. Hong, M. Kim, C. Lee, and S. Kim, "Head-mounted interface for intuitive vision control and continuous surgical operation in a surgical robot system," *Med. Biol. Eng. Comput.*, vol. 57, no. 3, pp. 601–614, Mar. 2019.
- [8] M. B. Schäfer, K. W. Stewart, and P. P. Pott, "Industrial robots for tele-operated surgery—A systematic review of existing approaches," *Current Directions Biomed. Eng.*, vol. 5, no. 1, pp. 153–156, Sep. 2019.
- [9] P. C. van der Sluis, S. van der Horst, A. M. May, C. Schippers, L. A. A. Brosens, H. C. A. Joore, C. C. Kroese, N. Haj Mohammad, S. Mook, F. P. Vleggaar, I. H. M. Borel Rinkes, J. P. Ruurda, and R. van Hillegersberg, "Robot-assisted minimally invasive thoracoscopic esophagectomy versus open transthoracic esophagectomy for resectable esophageal cancer: A randomized controlled trial," *Ann. Surgery*, vol. 269, no. 4, pp. 621–630, Apr. 2019.
- [10] C. Gruijthuijsen, G. Borghesan, D. Reynaerts, and E. V. Poorten, "A hybrid Active/Passive wrist approach for increasing virtual fixture stiffness in comanipulated robotic minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 3029–3036, Jul. 2019.
- [11] M. Fan, Y. Liu, D. He, X. Han, J. Zhao, F. Duan, B. Liu, and W. Tian, "Improved accuracy of cervical spinal surgery with robot-assisted screw insertion: A prospective, randomized, controlled study," *Spine*, vol. 45, no. 5, pp. 285–291, 2020.
- [12] F. Zhong, Y. Wang, Z. Wang, and Y.-H. Liu, "Dual-arm robotic needle insertion with active tissue deformation for autonomous suturing," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2669–2676, Jul. 2019.
- [13] F. Alambeigi, M. Bakhtiarinejad, S. Sefati, R. Hegeman, I. Iordachita, H. Khanuja, and M. Armand, "On the use of a continuum manipulator and a bendable medical screw for minimally invasive interventions in orthopedic surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 1, no. 1, pp. 14–21, Feb. 2019.
- [14] C. Forbrigger, A. Lim, O. Onaizah, S. Salmanipour, T. Looi, J. Drake, and E. D. Diller, "Cable-less, magnetically driven forceps for minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1202–1207, Apr. 2019.
- [15] M. Roizenblatt, A. T. Grupenmacher, R. Belfort Junior, M. Maia, and P. L. Gehlbach, "Robot-assisted tremor control for performance enhancement of retinal microsurgions," *Brit. J. Ophthalmol.*, vol. 103, no. 8, pp. 1195–1200, Aug. 2019.
- [16] Z. Wang, J. Wei, H. Liu, and Y. Gu, "Comparison analysis of robot-assisted computed tomography navigation system and manual freehand technique in orthopedic surgery," *J. Med. Imag. Health Informat.*, vol. 9, no. 2, pp. 349–353, Feb. 2019.
- [17] H. S. Park, J. Lee, H. Lee, K. Lee, S. Y. Song, and A. Toesca, "Development of robotic mastectomy using a single-port surgical robot system," *J. Breast Cancer*, vol. 23, no. 1, p. 107, 2020.
- [18] J. Fotouhi, T. Song, A. Mehrfard, G. Taylor, Q. Wang, F. Xian, A. Martin-Gomez, B. Fuerst, M. Armand, M. Unberath, and N. Navab, "Reflective-AR display: An interaction methodology for Virtual-to-Real alignment in medical robotics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2722–2729, Apr. 2020.
- [19] C. A. Nelson, M. A. Laribi, and S. Zeghloul, "Multi-robot system optimization based on redundant serial spherical mechanism for robotic minimally invasive surgery," *Robotica*, vol. 37, no. 7, pp. 1202–1213, Jul. 2019.
- [20] R. Lazzaro, B. Patton, P. Lee, J. Karp, E. Mihelis, S. Vatsia, and S. J. Scheinerman, "First series of minimally invasive, robot-assisted tracheobronchoplasty with mesh for severe tracheobronchomalacia," *J. Thoracic Cardiovascular Surgery*, vol. 157, no. 2, pp. 791–800, Feb. 2019.
- [21] D. Ji, T. H. Kang, S. Shim, and J. Hong, "Analysis of twist deformation in wire-driven continuum surgical robot," *Int. J. Control, Autom. Syst.*, vol. 18, no. 1, pp. 10–20, Jan. 2020.
- [22] G. Corrado, C. Zoccali, N. Salducca, A. Oddi, E. Vizza, and R. Biagini, "Anterior robotic approach in en-bloc sacrectomy: A preliminary experience," *J. Robotic Surgery*, vol. 13, no. 1, pp. 53–59, Feb. 2019.
- [23] F. Ficuciello, G. Tamburrini, A. Arezzo, L. Villani, and B. Siciliano, "Autonomy in surgical robots and its meaningful human control," *J. Paladyn Behav. Robot.*, vol. 10, no. 1, pp. 30–43, Jan. 2019.
- [24] H. Zhang, Z. Wang, Y. Zheng, F. Wang, Y. Geng, L.-Q. Chen, and Y. Wang, "Robotic-assisted simultaneous resection of a left-sided thymic neoplasm and a right-sided lower thoracic paravertebral neoplasm via the same ports and two docking steps: A case report," *J. Cardiothoracic Surgery*, vol. 14, no. 1, pp. 1–4, Dec. 2019.
- [25] R. Moccia, C. Iacono, B. Siciliano, and F. Ficuciello, "Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1650–1655, Apr. 2020.
- [26] B. Pokala, L. Flores, P. R. Armijo, V. Kothari, and D. Oleynikov, "Robot-assisted cholecystectomy is a safe but costly approach: A national database review," *Amer. J. Surgery*, vol. 218, no. 6, pp. 1213–1218, Dec. 2019.
- [27] H.-S. Liu, S.-J. Duan, F.-Z. Xin, Z. Zhang, X.-G. Wang, and S.-D. Liu, "Robot-assisted minimally-invasive internal fixation of pelvic ring injuries: A single-center experience," *Orthopaedic Surgery*, vol. 11, no. 1, pp. 42–51, Feb. 2019.
- [28] X. Wang, H. Lan, and K. Li, "Treatment of femoral neck fractures with cannulated screw invasive internal fixation assisted by orthopaedic surgery robot positioning system," *Orthopaedic Surgery*, vol. 11, no. 5, pp. 864–872, Oct. 2019.
- [29] E. R. Nagyné and T. Haidegger, "Robot-assisted minimally invasive surgical skill assessment—Manual and automated platforms," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 141–169, 2019.
- [30] G. Zhou, X. Chen, B. Niu, Y. Yan, F. Shao, Y. Fan, and Y. Wang, "Intraoperative localization of small pulmonary nodules to assist surgical resection: A novel approach using a surgical navigation puncture robot system," *Thoracic Cancer*, vol. 11, no. 1, pp. 72–81, Jan. 2020.
- [31] J. Kaouk, R. Valero, G. Sawczyn, and J. Garisto, "Extraperitoneal single-port robot-assisted radical prostatectomy: Initial experience and description of technique," *BJU Int.*, vol. 125, no. 1, pp. 182–189, Jan. 2020.
- [32] Y. Croissant, S. Zangos, M. H. Albrecht, K. Eichler, C. Schomerus, A. Spandorfer, U. J. Schoepf, T. J. Vogl, and C. Czerny, "Robot-assisted percutaneous placement of K-wires during minimally invasive interventions of the spine," *Minimally Invasive Therapy Allied Technol.*, vol. 28, no. 6, pp. 373–380, Nov. 2019.
- [33] F. P. Prete, A. Pezzolla, F. Prete, M. Testini, R. Marzaioli, A. Patriti, R. M. Jimenez-Rodriguez, A. Gurrado, and G. F. M. Strippoli, "Robotic versus laparoscopic minimally invasive surgery for rectal cancer: A systematic review and meta-analysis of randomized controlled trials," *Ann. Surgery*, vol. 267, no. 6, pp. 1034–1046, Jun. 2018.



**XIUMEI TIAN** was born in Shandong, China, in 1980. She received the bachelor's degree from Chongqing University, in 2004, and the master's degree from Tianjin University, in 2007. She is currently with Heze University. She has published two articles. Her research interests include control algorithm and motor control.



**YAN XU** was born in Henan, China, in 1983. She received the master's degree from Chongqing University, in 2007. She has been a Teacher with the North China University of Water Resources and Electric Power, since 2007, where she has been a Lecturer with the Department of Electric Power, since 2010. She has published five articles. She has conducted one research project and three teaching reform projects. Her research interests include wireless sensor networks and smart city.

• • •