Target Locating and Grabbing Algorithm for Robot Prosthesis Based on PSO-RBF

Jianbo Zhou¹, Chuanjiang Wang¹,²*, Xiujuan Sun², Kunpeng Wang¹,², Zhengkai Feng¹

¹Robot Research Center, Shandong University of Science and Technology, Qingdao, Shandong, 266590, China
²College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, Shandong, 266590, China
*Corresponding author’s e-mail: cxjwang@sdust.edu.cn

Abstract: Aiming at the target locating and grabbing problem for a five-degree-of-freedom intelligent robot prosthesis worn by the disabled, an algorithm is proposed. Firstly, the laser rangefinder and attitude sensor installed at the head of the wearer are used to simulate the motion of the human eye to realize the position of the target in the prosthesis workspace. Secondly, the laser ranging information is obtained by the 3D attitude sensor installed at the wearer's shoulder to obtain the position coordinates of the target relative to the prosthesis shoulder, so as to obtain the position coordinates of the target in the prosthesis workspace. Finally, the PSO-RBF neural network is used to train the samples to obtain the nonlinear mapping relationship between the position coordinates and the joint angle, thus predicting the joint angle and grabbing the target. The simulation and experimental results show that this method can locate the target accurately and calculate the angle of each joint of the prosthesis in real-time, so as to complete the action of target grabbing.

1. Introduction
Locating and grabbing the target is one of the main tasks of upper limb prosthesis. A method of locating the target position using a binocular vision camera is proposed by [1]. A biomimetic eye platform is designed by [2], and the coordinate measurement value of the target is obtained by using the binocular 3D measurement algorithm, and the rotation angles of the platform motors can be calculated to complete the real-time tracking of the moving target.

After obtaining the target coordinates, the common method is using inverse kinematics (IK) to achieve the angle of each joint, enabling the claw at the end of the prosthesis to reach the target and completing the grab of the target. The application of artificial neural network (ANN) to solve IK of serial manipulators has been studied for nearly 20 years. The performance of radial basis function (RBF) and Multi-layer Perceptron (MLP) neural networks in solving IK of six-degree-of-freedom manipulators was compared in [3]. A deep artificial neural network (DANN) using to inverse the five-degree-of-freedom manipulator was performed in [4]. An improved PSO-OSD algorithm was proposed by [5], and the accuracy of the algorithm was verified by experiment.

The prosthesis has a claw-self-balancing module, which can keep the claw mouth parallel to the ground when grabbing target [6], so there is no need to obtain the pose matrix of the claw. Applying neural network to solve the joint angle does not need a complete transformation matrix. In this paper,
PSO optimized RBF neural network is used to calculate the joint angle directly according to the position vector, so as to make the prosthesis claw reach the target position and complete the grabbing action, which saves the tedious calculation process of traditional method of solving IK.

2. Kinematics Model and Target Locating Method

2.1. Kinematics Model
The robot prosthesis physical prototype is shown in Figure 1, which has five degrees of freedom except the opening and closing of the claw. The coordinate frame of the prosthesis is also shown in Figure 1, where the coordinate system $O$-$XYZ$ defined at the shoulder is a fixed reference coordinate system and coincides with the origin of coordinate system $O_0$-$X_0Y_0Z_0$.

![Figure 1. Physical drawing and coordinate frames of robot prosthesis](image)

The D-H parameters of the robot prosthesis based on the standard D-H linkage parameter definition and the parameters of the kinematics coordinate system of the prosthesis are shown in Table 1.

| Links | $a$(mm) | $a$(deg) | $d$(mm) | (deg) |
|-------|----------|----------|---------|-------|
| 0     | 0        | 90       | 0       | 0     |
| 1     | 0        | 90       | 0       | -10−100 |
| 2     | 0        | -90      | 300     | -180−0 |
| 3     | 0        | 90       | 0       | -90−90 |
| 4     | 0        | 90       | 200     | 0−180 |
| 5     | 130      | 90       | 0       | 0−180 |

2.2. Method of Target Locating
(1) Sensors setting
As shown in Figure 2, the 3D attitude sensor 1 is installed in the middle of the head of the smart prosthesis wearer to detect the wearer's head posture, the 3D attitude sensor 2 is installed in the prosthesis shoulder to calibrate the zero-point posture, and the posture is regarded as the zero point when the wearer's head is straight and facing the front. The head motion angle of the wearer can be obtained by analyzing the data of the three position sensors. The laser rangefinder is mounted above the wearer's left ear for target confirmation and detection of target distance.

![Figure 2. Sensors setting](image)
(2) Coordinate system setting
As shown in Figure 2, coordinate system S is set up at the wearer’s shoulder, coordinate system N is set up at the wearer’s middle of the neck, coordinate system H is set up at the wearer’s middle of the head and the laser rangefinder coordinate system is E. In order to facilitate the description, make the following agreement: Rotation angles of one coordinate system around another coordinate system \(x, y, z\) axis is recorded as \(\alpha_i, \beta_i, \gamma_i\), respectively. The origin of one coordinate system is expressed as \(O_i(x_i, y_i, z_i)\) (\(i = S, N, H, E\)) in another coordinate system. A transformation matrix of one coordinate system relative to another is defined as T.

(3) Target location calculation
The wearer adjusts the posture of head to make the laser point emitted by the laser rangefinder aim at the target. At this time, the linear distance between the laser rangefinder and the target is \(d\), then the position of the target in the coordinate system E can be expressed as:

\[
^{E}M = \begin{bmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]  

(1)

According to the principle of coordinate transformation, the real-time coordinate \(p\) of the target in prosthetic workspace can be expressed as:

\[
^{S}p = ^{S}T \cdot ^{N}T \cdot ^{H}T \cdot ^{E}M
\]  

(2)

Among them:

\[
^{S}T = \begin{bmatrix} 
\cos(\beta_i) & 0 & \sin(\beta_i) \\
0 & 1 & 0 \\
\sin(\beta_i) & 0 & \cos(\beta_i)
\end{bmatrix}
\]  

(3)

\[
^{N}T = \begin{bmatrix} 
\cos(\alpha_s) & -\sin(\alpha_s) & 0 \\
\sin(\alpha_s) & \cos(\alpha_s) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

(4)

\[
^{H}T = \begin{bmatrix} 
\cos(\alpha_h) & -\sin(\alpha_h) & 0 \\
\sin(\alpha_h) & \cos(\alpha_h) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

(5)

3. Algorithm for Solving Joint Angles

3.1. PSO-RBF Network Model
Radial Basis Function Neural Network (RBFNN) includes input layer, hidden layer and output layer. Figure 3 represents an RBF network structure for solving 5 joint angles. The neuron activation function of the hidden layer is a radial basis function. The mapping relationship between each hidden layer node
containing a central vector \( c \). The target position vector \( p \) and the output of the hidden layer can be indicated by the activation function \( h_j(t) \):

\[
h_j(t) = \exp\left\{ \frac{|p(t) - c_j(t)|}{2k_j} \right\}, \quad j = 1, \ldots, m
\]

\( k_j \) represents the width of the Gaussian kernel function, \( m \) is the number of hidden layer nodes. The output of the network is represented by (7):

\[
\theta(t) = \sum_{j=1}^{n} w_j h_j(t), i = 1, \ldots, n
\]

Where \( w \) is the output layer weight and \( n \) is the number of output layer nodes.

Figure 3. RBF neural network for computation of joint angle

Particle swarm optimization (PSO) algorithm originates from a study which is about the foraging behavior of birds. Each particle in D-dimensional space represents a search individual. The current position of the particle is a candidate solution to the optimization problem. And all particles share position information. The particle position and velocity are constantly updated by (8) and (9) until the optimal solution that meets the requirements is obtained.

\[
v_{id}^{k+1} = \omega v_{id}^k + c_1 \times \text{rand}^k(0,1) \times (P_{bid}^k - x_{id}^k) + c_2 \times \text{rand}^k(0,1) \times (G_{bid}^k - x_{id}^k)
\]

\[
x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}
\]

\( \omega \) is an inertial factor, and its value is nonnegative. The value of \( \omega \) affects the global and local optimization ability. \( c_1 \) and \( c_2 \) are acceleration constants, and they are set to 2 in most cases. \( \text{rand} (0,1) \) represents a random number uniformly distributed on the interval [0,1]. \( P_{bid} \) represents the \( d \)-th dimension of the individual extremum of the \( i \)-th variable, \( G_{bid} \) represents the \( d \)-th dimension of the global optimal solution. \( v_{id} \) is the velocity vector of the \( i \)-th particle of the \( d \)-dimension in the particle swarm. \( x_{id} \) is the position vector of the \( i \)-th particle of the \( d \)-dimension in the particle swarm.

3.2. Algorithm for Solving Joint Angles

\( \{c, k, w\} \) is regarded as the optimal object of the PSO algorithm, where \( c \) is the centre of the base function of the RBF neural network, \( k \) is the hidden unit width, and \( w \) is the weight of the hidden layer to the output layer. These three parameters are considered as moving particles, and the neural network is established by the optimal solution which is calculated by the PSO algorithm. This method can improve the performance of neural network and reduce the training error [7].

Position vector \( p=[X,Y,Z] \) of the target in prosthesis workspace is set as network input, prosthesis joint rotation angle \( \theta=[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5] \) is set as the network output, the joint angle can be solved by the nonlinear mapping relationship between the input and output which are obtained after training the PSO-RBF neural network. The steps are as follows:

Step 1: Initializing the default parameters of particle swarm and neural network, the results that the particle swarm dimension \( D \) is equal to 54 and the number of network hidden layers \( n \) is equal to 6 can be identified based on the input and sample dimension.

Step 2: Selecting the MSE of the neural network as the fitness function and calculating the fitness of the particle according to (10).
\[ \text{FitFun}_i = \frac{1}{1 + \frac{1}{N} \sum_{j=1}^{N} \left( y_{ij} - y_{ij}' \right)^2} \]  

(10)

Where \( N \) is the number of training samples, \( m \) is the number of output layer nodes, \( y_{ij} \) and \( y_{ij}' \) are the actual value and expected value of the \( d \)-th sample at the \( i \)-th node in the output layer, respectively.

Step 3: Updating individual particle extremum \( Pb \) based on the value of \( \text{FitFun}_i \) calculated in step 2.

Step 4: Updating the global optimal extremum \( Gb \) of the population.

Step 5: Determining whether the termination condition is satisfied. If not, return to step 2. If satisfied, output the global optimal solution \( Gb \). This is the neural network optimal kernel function parameter.

Step 6: Using the optimized parameters to establish RBF neural network, setting the position vector \( p \) as network input, and setting the joint angle \( \theta \) as network output, obtaining the mapping relationship between input and output after training, and storing the results in the established function PSORBF_NET.

Step 7: Input the target coordinate \( p \) in the PSORBF_NET, and achieving the joint angle \( \theta \).

4. Experiment and Results

4.1. Samples Collection

In order to avoid overfitting of the training model on the local path, a random trajectory of the claw at the end of the robot prosthesis should be generated instead of the predetermined trajectory [8]. The moving head laser point generates multiple trajectories, each trajectory is composed of multiple path points. The position coordinates \( p \) of each point and the corresponding joint rotation angle \( \theta \) is a set of sample. 10000 sets of samples are recorded by the software. Taking the large difference in the original data range into account, the collected data should be normalized between [0,1] by the mapminmax function in MATLAB. Then 75% of the total dataset is used for training and 25% is used for validation.

4.2. PSO-RBF Network Training

Using MATLAB to establish kinematics model and PSO-RBF network model, assuming that particle swarm dimension \( D \) is equal to 54, \( c_1 \) and \( c_2 \) are equal to 2, and the position and velocity range is [-1,1]. The number of RBF network input and output layers are 3 and 6, respectively, and the training termination error condition is 0.0001. The networks performance is shown in Table 2.

| Model      | MSE  |
|------------|------|
| BP         | 0.0023 |
| RBF        | 0.0017 |
| PSO-RBF    | 0.0008 |

4.3. Simulation and Experiment

The joint angle \( \theta_0 \) of the initial position of the prosthesis [9] is [0,90°,0,-90°,-90°], and the selected path of the claw is composed of \( P_0,P_1,P_2 \). Moreover, the \( p_r=[0,0,630] \) is the position vector of the claw when the prosthesis is in the initial position. The joint angles of the three nodes are solved by the PSO-RBF algorithm, the three-layer BP neural network and the conventional RBF network with \( \text{spread} = 0.2 \), respectively. The results are shown in Table 3.

| Nodes | Target location | Joint Expected | PSO-RBF | BP | RBF |
|-------|-----------------|----------------|---------|----|-----|
| \( P_1 \) | X -247.6464 | \( \theta_1 \) -16.641 | -16.891 | -16.088 | -17.110 |
|       | Y 313.0898 | \( \theta_2 \) 92.932 | 92.892 | 92.485 | 92.542 |
|       | Z 372.7295 | \( \theta_3 \) 43.066 | 43.017 | 43.823 | 43.573 |
|       | \( \theta_1 \) -149.728 | -149.808 | -150.281 | -150.811 |
|       | \( \theta_2 \) -101.675 | -101.594 | -102.581 | -101.908 |
| \( P_2 \) | X 14.8115 | \( \theta_1 \) -48.497 | -48.426 | -49.367 | -48.976 |
|       | Y 288.0080 | \( \theta_2 \) 157.004 | 157.089 | 157.994 | 157.585 |
On the basis of the calculated joint angle, assuming that the sampling interval is 0.2s, and the time is 10s. Drawing the trajectory of the claw using MATLAB RVC toolbox [10]. The simulation trajectory of the claw shown in the Figure 4 shows that the trajectory predicted by the PSO-RBF network can better fit the expected trajectory relative to the BP network and the conventional RBF network.

|   | Z  | θ₁ | θ₂ | θ₃ |
|---|----|----|----|----|
|   | -59.9571 | 63.978 | 63.802 | 63.057 | 64.699 |
|   | -28.346 | -28.254 | -29.083 | -28.604 |
|   | -74.439 | -74.321 | -75.217 | -74.918 |

In order to verify the accuracy and reliability of the proposed algorithm in practical application, multiple experiments on locating and grabbing targets are performed on a prototype prosthesis. Selecting the position of the test point as shown in Figure 5(a), using the algorithm proposed in section 3.2 to calculate the coordinate of the test point in the prosthesis workspace, and inputting the coordinate into the trained RBF network, BP network and PSO-RBF network respectively to obtain three groups of joint angle solutions, the experimental results of RBF network, BP network and PSO-RBF network are shown in Figure 5(a) from left to right. It can be seen that the deviations between the centre point of the claw and the test point is obvious in x-axis and the deviation of PSO-RBF network is the smallest. A locating and grabbing action on the target completed by the robot prosthesis is shown in Figure 5(b).
5. Conclusions
This paper proposes an algorithm for target location and grasping of the prosthesis workspace:

1. Using 3D attitude sensor and laser rangefinder to determine the position coordinates of the target in the prosthesis workspace.
2. The obtained position coordinates are taken as input, and the trained PSO-RBF neural network is used to predict the angles of the prosthesis joint, this enables the prosthesis claw to reach the target position and achieve the grab of the target.

The results of the simulation and experiment show that PSO-RBF network is superior to the conventional RBF network and the BP network. Furthermore, the MSE of this method is 52.9% and 65.2% lower than the conventional RBF network algorithm and the BP network respectively.

Acknowledgements
This research was financially supported by Key Research and Development Plan of Shandong Province (Grant No. 2016GSF201197).

References
[1] Fan Binghui, Liu Guigui, et al. Research on positioning system of senior and disable people aid robot based on stereo vision[J]. Modern Electronics Technique, 2017, 40(02):48-52 (in Chinese).
[2] Li Zhen, Fan Binghui, et al. Locating and Tracking Algorithm of Biomimetic Eye Based on Backpropagation Neural Network[J]. Robot, 2017, 39(01):63-69 (in Chinese).
[3] Shital S, Chiddarwar, N. Ramesh Babu. Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach[J]. Engineering Applications of Artificial Intelligence, 2010, 23(7):1083–1092
[4] Shubham Kamlesh Shah, Ruby Mishra, Lala Samprit Ray. Solution and validation of inverse kinematics using Deep Artificial neural network[J]. Materials Today: Proceedings, 2020, 26(2):1250–1254.
[5] Vahid Fathi, Gholam Ali Montazer. An improvement in RBF learning algorithm based on PSO for real time applications, Neurocomputing, 2013, 111(2):169-176.
[6] Wang Chuanjiang, Wang Dong, et al. Design of hand self-balancing system for the forearm prosthesis based on quaternion complementary filtering algorithm[J]. Robot Technique and Application, 2018(01):34-37 (in Chinese).
[7] Wang Zhongmin, Yang Dongfang, et al. Research on Neural Network Prediction of Power Transmission and Transformation Project Cost Based on GA-RBF and PSO-RBF[J]. Applied Mechanics and Materials, 2014, 644-650:2526-2531.
[8] Hailin Ren, Pinhas Ben-Tzvi. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. Robotics and Autonomous System, 2020, 124:103386.
[9] Chen Chen. Mechanical structure design and key technology research of robot prosthesis[D] Shandong: Shandong University of Science and Technology, 2020:42-43(in Chinese).
[10] Peter Corke. Robotics, Vision and Control: Fundamental Algorithm in MATLAB[M]. Beijing: publishing house of electronic industry,2016:148-15.