

## A Trajectory Planning Algorithm for Medical Manipulators Based on Adaptive Particle Swarm Optimization and Fuzzy Neural Network

Fengmei Gao\*, Pan Wu

School of Smart Health, Chongqing College of Electronic Engineering, Chongqing 401331, China

Corresponding Author Email: [gaofengmei@cqcet.edu.cn](mailto:gaofengmei@cqcet.edu.cn)

<https://doi.org/10.18280/jesa.530314>

### ABSTRACT

**Received:** 18 March 2020

**Accepted:** 1 June 2020

#### Keywords:

*particle swarm optimization (PSO) algorithm, medical manipulator, jitter suppression, fuzzy neural network (FNN)*

With the development of smart healthcare, surgical and rehabilitation robots have permeated into daily medical operations. This raises concerns over the trajectory planning of medical manipulators. Based on particle swarm optimization (PSO) algorithm and fuzzy neural network (FNN), this paper puts forward a trajectory planning algorithm for medical manipulators, which ensures that the target medical manipulator can suppress the residual jitter at the end, while meeting the requirements of high precision, flexible operation, and disturbance resistance. Specifically, a kinetic model was constructed for a medical manipulator of multi-degrees-of-freedom (DOFs) through position and posture transforms, and used to construct an FNN for trajectory planning. To suppress the jitter at the end, an adaptive PSO algorithm was designed, and combined with the FNN into a trajectory planning algorithm called PSO neural network (PSO-NN) algorithm. Finally, the proposed algorithm was proved effective through experiments. The research results provide the reference for applying PSO algorithm and FNN in other fields.

## 1. INTRODUCTION

With the rapid progress of artificial intelligence (AI) and computer technology, industrial robots have been gradually applied in various industries. The technology of industrial robots is increasing mature, as evidenced by the integration of multiple joints, central processing units (CPUs), and sensors [1-6].

Recently, medical robots and other medical equipment have attracted much attention. To assist doctors in surgeries, medical manipulators must be highly accurate and flexible, and good at jitter suppression [7-9], ensuring the stability and continuity of velocity and acceleration. Before the surgery, the manipulator trajectory must be optimized to minimize the impact on each joint, and to precisely reach the designated positions [10-12].

Currently, the trajectories of applied manipulators are mainly planned through obstacle avoidance by artificial potential field (APF) method, trajectory length calculation by ant colony algorithm, and accurate graph search based on machine vision [13-15]. Mohamed et al. [16] found the optimal solution to joint variables with distributed AI, and solved the reverse motion of a seven-degrees-of-freedom (DOFs) manipulator. Through nonuniform B-spline interpolation, Annisa et al. [17] optimized manipulator trajectory under the constraints of acceleration and torque, and thereby improved the real-time control effect of the manipulator. Using adaptive impedance, Annisa et al. [18] offset the effect of initial parameter values on manipulator control system, and increased the accuracy and velocity of repetitive manipulator motions through iterative learning. Jamali et al. [19] set up an object space coordinate system for a flexible robot platform, constructed a neural network (NN)-

based impedance control strategy for different unknown disturbances, and expanded the applicable scope of the robot by adjusting the impedance.

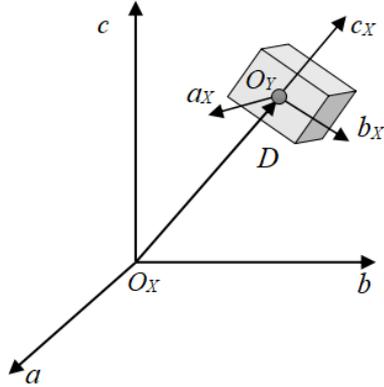
To meet the needs and effects of surgeries, medical manipulators generally have multiple DOFs. The large number of DOFs poses a key difficulty in motion control and trajectory planning of medical manipulators. Therefore, many experts and scholars have attempted to balance the DOF, accuracy, flexibility, and jitter suppression of medical manipulators [20-23]. Based on the topology of constrained space, Wilkening et al. [24] modelled the DOFs of a flexible medical manipulator, and designed the structure of a variable rigidity medical manipulator, capable of multi-DOF motions and effective lock-up. Based on the real-time video images captured by the Vision Development Module of LabVIEW, Sefati et al. [25] carried out three-dimensional (3D) observation of the manipulator from multiple angles, and realized the real-time observation and control of the medical process.

Based on particle swarm optimization (PSO) algorithm and fuzzy neural network (FNN), this paper proposes a trajectory planning algorithm for medical manipulators called PSO neural network (PSO-NN) algorithm. The PSO-NN enables the target medical manipulator to adapt to the changes in system disturbances and suppress the residual jitter at the end, while meeting the requirements of high precision and flexible operation. Firstly, a kinetic model was built for a n-DOF medical manipulator through position and posture transforms, and used to construct an FNN for trajectory planning. To suppress the jitter at the end, an adaptive PSO algorithm was designed, and combined with the FNN to plan the trajectories of the medical manipulator. The proposed method was proved effective through experiments.

## 2. KINETIC MODEL OF MEDICAL MANIPULATOR

The DOFs of a manipulator refer to the number of independent motion parameters that must be given to describe the manipulator's motion, turning or rotation. This number is often the same as that of driving mechanisms (hydraulic or electric motors). Each DOF corresponds to a mechanical joint.

The workspace range of the manipulator depends on two factors: the length of each link and the configuration space of each joint. Hence, a reference CS  $X$  and the CS of the rigid link  $Y$  were set up for the manipulator workspace. Let vector  $D=[abc]^T$  be any point in the workspace (See Figure 1).



**Figure 1.** The reference CS and rigid link CS

Then, the direction of the rigid link relative to CS  $X$  can be expressed as:

$$P_{X-Y} = \begin{bmatrix} X_{a_Y} & X_{b_Y} & X_{c_Y} \end{bmatrix} = \begin{bmatrix} u_a & v_a & w_a \\ u_b & v_b & w_b \\ u_c & v_c & w_c \end{bmatrix} \quad (1)$$

where,  $P_{X-Y}$  is the transform matrix from CS  $Y$  to CS  $X$ . The matrix consists of the direction cosines of the three unit principal vectors  $X_{a_Y}$ ,  $X_{b_Y}$ , and  $X_{c_Y}$  relative to the CS  $X$ .

The position vector  $[abc]$  can be combined with formula (1) to illustrate the position and posture of the rigid link mapped from CS  $Y$  to CS  $X$ :

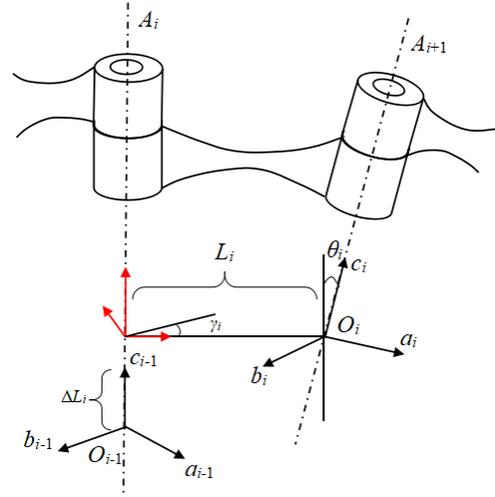
$$H_{X-Y} = \begin{bmatrix} P_{X-Y} & D \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} u_a & v_a & w_a & a \\ u_b & v_b & w_b & b \\ u_c & v_c & w_c & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The medical manipulator is a multi-joint multi-DOF device with both flexible and rigid links. The transform matrix  $W_i$  between the two kinds of links can be described as:

$$W_i = R(b, \gamma_i) * T(0, 0, \Delta L_i) * T(L_i, 0, 0) * R(a, \theta_i) \quad (3)$$

where,  $L_i$  is the length of the common perpendicular between the axis of the  $i$ -th joint  $A_i$  and that of the  $i+1$ -th joint  $A_{i+1}$ ;  $\theta_i$  is the angle between the axis of the  $i+1$ -th joint  $A_{i+1}$  and the plane  $F$  made of the axis of the  $i$ -th joint  $A_i$  and the common perpendicular  $L_i$ ;  $\Delta L_i$  is the distance between the two common perpendiculars  $L_i$  and  $L_{i-1}$  of the  $i$ -th joint  $A_i$ ;  $\gamma_i$  is the angle between the projections of common perpendiculars  $L_i$  and  $L_{i-1}$

on the plane with the axis of the axis of the  $i$ -th joint  $A_i$  as the normal.



**Figure 2.** The position and posture transforms of the link

As shown in Figure 2, the positions and angles of the link between the two joints can be transformed by  $W_i$  through two rotations (by  $\theta_i$  about  $x$ -axis and  $\gamma_i$  about  $z$ -axis) and two translations (by  $L_i$  and  $\gamma_i$ ).

The cosine form of  $W_i$  can be expressed as:

$$W_i = \begin{bmatrix} \cos \gamma_i & -\sin \gamma_i \cos \theta_i & \sin \gamma_i \sin \theta_i & L_i \cos \gamma_i \\ \sin \gamma_i & \cos \gamma_i \cos \theta_i & -\cos \gamma_i \sin \theta_i & L_i \sin \gamma_i \\ 0 & \sin \theta_i & \cos \theta_i & \Delta L_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

If the rigid connection position of CS  $Y$  is moved from the original end to the other end, formulas (3) and (4) can be transformed into:

$$W_i = T(L_i, 0, 0) * R(a, \theta_i) * T(0, 0, \Delta L_i) * R(b, \gamma_i) \\ = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & L_{i-1} \\ \sin \gamma_i \cos \theta_{i-1} & \cos \gamma_i \cos \theta_{i-1} & -\sin \theta_{i-1} & -\Delta L_i \sin \theta_{i-1} \\ \sin \gamma_i \sin \theta_{i-1} & \cos \gamma_i \sin \theta_{i-1} & \cos \theta_{i-1} & -\Delta L_i \cos \theta_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

To determine the joint parameters, position, and speed of the multi-link  $n$ -DOF medical manipulator, the end position and posture of each link can be solved by the transform matrix  $W_i$  of each link:

$$W = W_1 \cdot W_2 \dots W_n \quad (6)$$

Substituting the total kinetic energy and total potential energy of the manipulator into the Lagrangian equation, the kinetic model of the manipulator can be established as:

$$\sum_{i=1}^n J_i(\theta) \ddot{\theta}_i + C_i(\theta, \dot{\theta}) + G_i(\theta) = \tau \quad (7)$$

where,  $J_i$ ,  $C_i$ , and  $G_i$  are the rotational inertia matrix, centripetal matrix, and gravity matrix of the  $i$ -th joint of the robotic arm, respectively;  $\tau_i$  is the driving torque vector of the  $i$ -th joint of the  $i$ -th joint:

$$J_i(\theta) = \begin{bmatrix} (m_i + m_{i+1})L_{i+1}^2 + m_{i+1}L_{i+1}^2 + 2m_{i+1}L_iL_{i+1}\cos\theta_{i+1} \\ m_{i+1}L_{i+1}^2 + m_{i+1}L_iL_{i+1}\cos\theta_{i+1} \\ m_{i+1}L_{i+1}^2 + m_{i+1}L_iL_{i+1}\cos\theta_{i+1} \\ m_{i+1}L_{i+1}^2 \end{bmatrix} \quad (8)$$

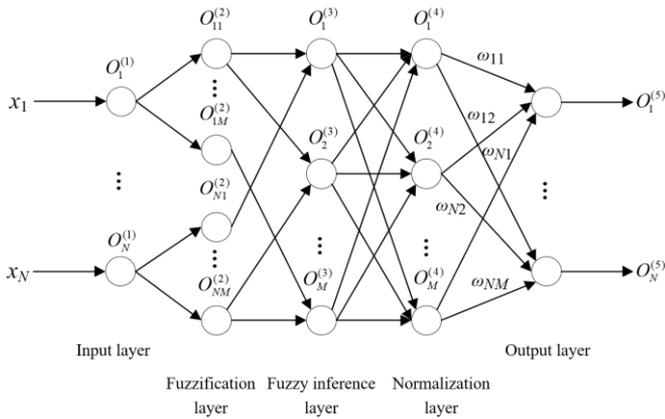
$$C_i(\theta, \dot{\theta}) = \begin{bmatrix} -m_{i+1}L_iL_{i+1}(2\dot{\theta}_i\dot{\theta}_{i+1} + \dot{\theta}_{i+1})\sin\theta_{i+1} \\ m_{i+1}L_iL_{i+1}\dot{\theta}_i\sin\theta_{i+1} \end{bmatrix} \quad (9)$$

$$G_i(\theta) = \begin{bmatrix} m_i g L_i \cos\theta_i + m_{i+1} g L_{i+1} \cos\theta_{i+1} \\ m_{i+1} g L_{i+1} \cos\theta_{i+1} \end{bmatrix} \quad (10)$$

where,  $m_i$ , and  $m_{i+1}$  are the mass of the two links connected to the  $i$ -th joint, respectively;  $g$  is the acceleration of gravity.

### 3. FNN CONSTRUCTION

To improve the trajectory control accuracy of the medical manipulator, this paper sets up an FNN based on the above kinetic model. As shown in Figure 3, the FNN consists of five layers, namely, an input layer, a fuzzification layer, a fuzzy inference layer, a normalization layer, and an output layer.



**Figure 3.** The structure of the FNN

The position and posture signals of each joint  $O_i^{(1)}=x_i$ ,  $i=1, 2, \dots, N$  were imported to the input layer.

With  $M$  fuzzy control rules, the fuzzification layer performs fuzzification of the signals with a Gaussian membership function:

$$O_{ij}^{(2)} = \exp\left[-\frac{(x_i - a_{ij})^2}{\sigma_{ij}^2}\right] \quad (11)$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, M$

where,  $a_{ij}$  and  $\sigma_{ij}$  are the center and width of the linguistic variable set of the  $j$ -th membership function of the position and posture signals of the  $i$ -th element, respectively.

In the fuzzy inference layer, fuzzy inference is carried out by the fuzzy control rule of weighted multiplication. The output of the fuzzy inference layer can be expressed as:

$$O_j^{(3)} = \prod_{i=1}^N O_{ij}^{(2)}(x_i) \quad j = 1, 2, \dots, M \quad (12)$$

The output of the normalization layer can be expressed as:

$$O_j^{(4)} = \frac{O_j^{(3)}}{\sum_{j=1}^M O_j^{(3)}} \quad (13)$$

The output layer performs defuzzification of the normalized output of the fuzzy inference layer, producing the output of the entire FNN.

Let  $O_e=e_i$ ,  $i=1, 2, \dots, N$  be the error signals of the position and posture of each joint, and  $\omega_j$  be the weight coefficient of the output layer, reflecting the coupling effect of each output of the FNN. Then, the driving signal from the driving system of each joint can be expressed as:

$$O_i^{(5)} = \sum_{j=1}^M \omega_j O_j^{(4)} \quad (14)$$

Let  $O_d$  be the desired driving signal of the FNN. Then, the target error of the FNN can be expressed as:

$$E = \frac{1}{2} \sum_{i=1}^N (O_i^{(5)} - O_{di})^2 \quad (15)$$

Let  $o_{ij}$  be the input of the  $s$ -th fuzzy control rule. Then, there exists a  $R_{ij}$  that satisfies:

$$R_{ij} = \sum_{j=1, j \neq i}^N o_{ij} j \quad (16)$$

Then, the backpropagation errors can be described as:

$$\varepsilon_{ij}^{(2)} = \sum_{s=1}^P \varepsilon_j^{(3)} R_{ij} \exp\left[-\frac{(x_i - a_{ij})^2}{\sigma_{ij}^2}\right] \quad (17)$$

$$\varepsilon_j^{(3)} = \frac{\left[ \varepsilon_j^{(4)} \sum_{i=1, i \neq j}^P O_j^{(3)} - \sum_{s=1, s \neq j}^P \varepsilon_s^{(4)} O_s^{(3)} \right]}{\left( \sum_{i=1}^P O_j^{(3)} \right)^2} \quad (18)$$

$$\varepsilon_j^{(4)} = \sum_{i=1}^N -\varepsilon_j^{(5)} \omega_j \quad (19)$$

$$\varepsilon_i^{(5)} = \frac{\partial E}{\partial O_i^{(5)}} = \sum_{i=1}^P -e_i \frac{\partial O_{di}}{\partial O_i^{(5)}} \quad (20)$$

Hence, the first-order error gradients can be solved by:

$$\frac{\partial E}{\partial a_{ij}} = -\frac{2\varepsilon_{ij}^{(2)}(x_i - a_{ij})}{\sigma_{ij}^2} \quad (21)$$

$$\frac{\partial E}{\partial \sigma_{ij}} = -\frac{2\varepsilon_{ij}^{(2)}(x_i - a_{ij})^2}{\sigma_{ij}^3} \quad (22)$$

$$\frac{\partial E}{\partial \omega_j} = -(O_i^{(5)} - O_{di})O_j^{(4)} \quad (23)$$

Let  $\tau > 0$  be the learning rate. During the learning of the FNN, the center  $a_{ij}$  and width  $b_{ij}$  of the linguistic variable set, and the weight coefficient  $\omega_j$  can be updated by the following rules:

$$a_{ij}(t+1) = a_{ij}(t) - \tau \frac{\partial E}{\partial a_{ij}(t)} \quad (24)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \tau \frac{\partial E}{\partial \sigma_{ij}(t)} \quad (25)$$

$$\omega_j(t+1) = \omega_j(t) - \tau \frac{\partial E}{\partial \omega_j(t)} \quad (26)$$

The advantage of the FNN relies in the ability to approximate the ideal function at any accuracy. On this basis, the driving signal of the medical manipulator can be expressed as the correlation equations between the target error and the FNN parameters  $a_{ij}$ ,  $b_{ij}$ , and  $\omega_j$ . Through network learning, the membership functions and parameter weights of the FNN can be adjusted automatically, such that the position and posture of the medical manipulator can be controlled accurately, without needing the positioning information of the system.

#### 4. DESIGN OF ADAPTIVE PSO ALGORITHM

For the medical manipulator, the residual jitter at the end needs to be suppressed, aiming to reduce the jitter-induced displacement at the end to zero. For this purpose, this paper designs an adaptive PSO algorithm to minimize the moment of driving force and the end displacement induced by residual jitter. The designed algorithm solves the target error function, and feeds back the result to the FNN, making the FNN self-adaptive. In this way, the FNN can adaptively adjust its parameters, position, and posture according to the driving signal changes of each joint, while effectively suppressing the residual jitter at the end.

In the traditional PSO algorithm, the velocity  $v_{id}$  and position  $x_{id}$  of the  $i$ -th particle are updated by formulas (27) and (28), respectively, in the  $d$ -dimensional space:

$$v_{id}(t+1) = \nu v_{id}(t) + c_1 \text{rand}_1 [p_{id} - x_{id}(t)] \quad (27)$$

$$+ c_2 \text{rand}_2 [p_{gd} - x_{id}(t)]$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (28)$$

where,  $p_{id}$  and  $p_{gd}$  are the currently best-known positions of the  $i$ -th particle and the swarm, respectively;  $c_1$  and  $c_2$  are two nonnegative acceleration factors;  $\text{rand}_1$  and  $\text{rand}_2$  are two random numbers between zero and one;  $\nu$  is the inertia weight.

The inertia weight  $\nu$  is positively correlated with the global search ability and negatively correlated with the local search ability of the PSO algorithm. In the early phase of optimization, the inertia weight should be increased to ensure the global search ability; in the latter phase, the inertia weight should be reduced to speed up the convergence.

Let particle fitness  $fit$  be the target error of the FNN. To strike a balance between global and local search abilities, the adaptive inertia weight was introduced as:

$$\nu = \begin{cases} (1 - \frac{t}{t_{max}})\nu_{max} + (1 + \frac{t}{t_{max}})\nu_{min}, & fit \leq fit_{avg} \\ (1 - \frac{t}{t_{max}})\nu_{max} + \frac{t}{t_{max}}\nu_{min}, & fit > fit_{avg} \end{cases} \quad (29)$$

where,  $t$  is the number of iterations;  $t_{max}$  is the maximum number of iterations;  $\nu_{max}$  and  $\nu_{min}$  are the maximum and minimum inertia weights, respectively;  $fit_{avg}$  is the average fitness of the swarm.

To model the modal shape of each link, the end displacement of an  $n$ -DOF medical manipulator can be expressed as:

$$\Delta s(t) = \sum_{i=1}^n D_i(t)W_i(L_i) \quad (30)$$

where,  $D_i$  is the coordinates of the end of each link;  $W_i$  is the transform matrix of each link. To make  $\Delta s(t) \rightarrow 0$ , this paper presents a trajectory planning algorithm called the PSONN. The main steps of the PSONN algorithm are as follows:

Step 1. Let  $t=1$ , and randomly initialize the positions  $\{x_1(t), x_2(t), \dots, x_K(t)\}$  and velocities  $\{v_1(t), v_2(t), \dots, v_K(t)\}$  of  $K$   $D$ -dimensional particles.

Step 2. Derive the relationship between joint motions and modal coordinates of the medical manipulator by formulas (6) and (7), respectively, producing the modal coordinates of the manipulator under the current joint trajectory; Solve the displacement variable at the end by formula (25); Calculate the fitness of each particle by formula (15).

Step 3. When  $t=1$ , take the local extreme value of particles as the global extreme value; when  $t > 1$ , replace local optimal value by the smallest extreme value of the new swarm.

Step 4. When  $t > 1$ , replace the global optimal value by the smallest extreme value of the swarms of all  $t$  iterations.

Step 5. Update the position and velocity of each particle by formulas (22) and (23), respectively.

Step 6. Repeat the above steps until the maximum number  $t_{max}$  of iterations is reached or the acceptable optimal value is found.

#### 5. EXPERIMENTS AND RESULTS ANALYSIS

To verify its performance, the proposed PSONN algorithm was applied to the trajectory planning of a 3DOF medical manipulator. A total of 15 fuzzy control rules were configured for the FNN. The parameters of the medical manipulator system were set as  $L_1=L_2=140\text{mm}$ ,  $L_3=100\text{mm}$ ,  $m_1=m_2=0.46\text{kg}$ , and  $m_3=0.38\text{kg}$ .

Figure 4 shows the variation curve of the optimal target error during the adaptive PSO. It can be seen that the curve was monotonously decreasing until the optimal value was

found or the maximum number  $t_{\max}$  of iterations was reached. In actual operation, the desired optimal value was obtained after about 1,500 iterations; after that, the target error remained basically unchanged.

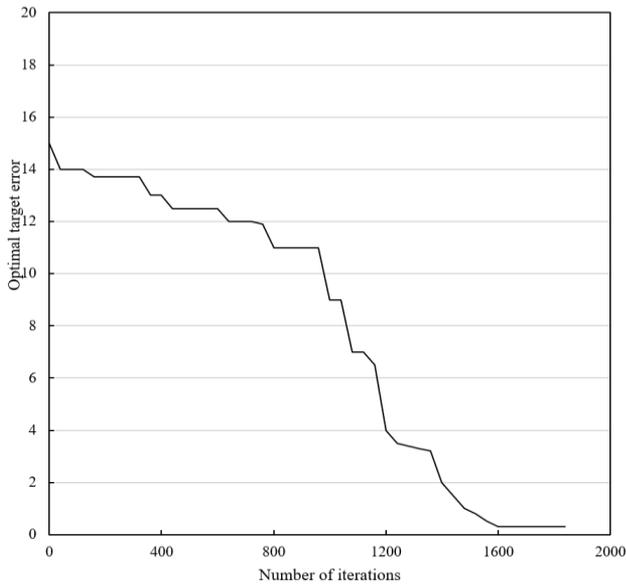


Figure 4. The variation curve of the optimal target error

Next, the proposed PSOINN algorithm was compared with the traditional proportional-integral-derivative (PID) control algorithm. Figures 5 and 6 compare the two algorithms in the tracking errors of the angular displacement and displacement at the end of the first and second joints, respectively.

It can be seen that, when the target error of trajectory planning was the same, the PSOINN algorithm reduced the tracking error of angular displacement at the end of the first and second joints by 38.79% and 30.48%, respectively, from that of the PID control algorithm; similarly, the PSOINN algorithm reduced the tracking error of displacement at the end of the first and second joints by 41.34% and 35.42%, respectively. Overall, our algorithm has a much smaller error than the PID control algorithm in tracking the manipulator trajectory.

Comparing Figures 5(a) and 5(b), both the FNN and PSOINN controlled the peak tracking error of angular error within  $\pm 1.75$ . This partially demonstrates the suppression effect of our algorithm on end jitter, which benefits the smooth operation of medical manipulators.

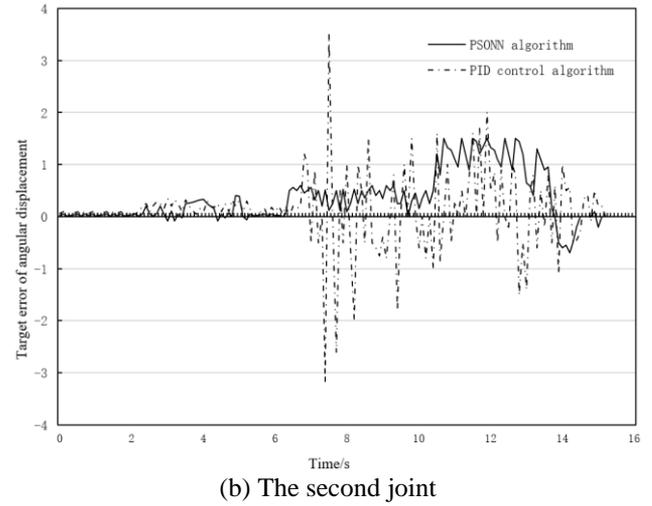


Figure 5. The comparison of the tracking errors of the angular displacement at the end of two joints

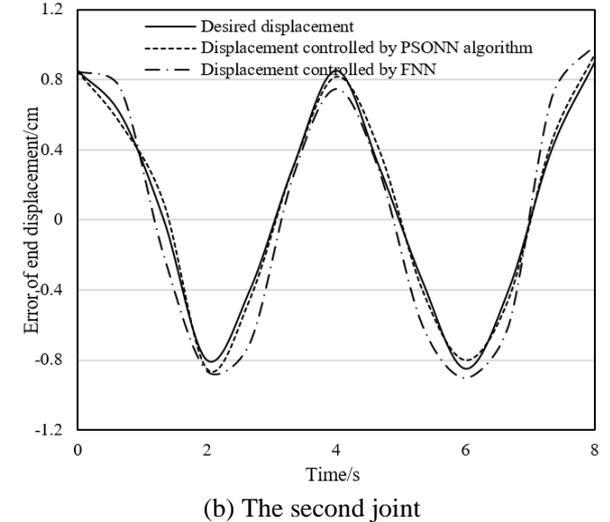
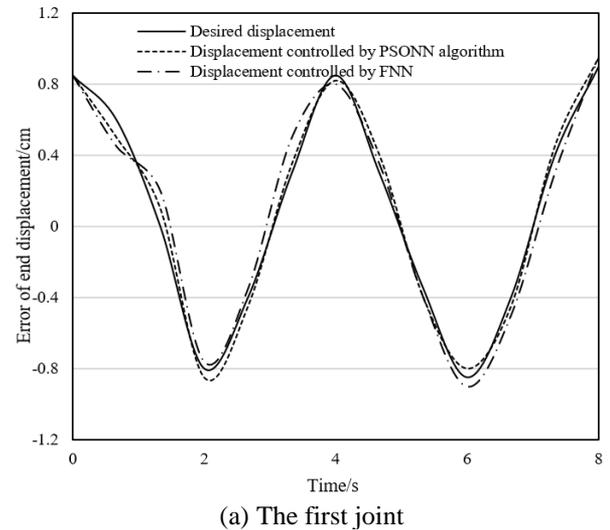
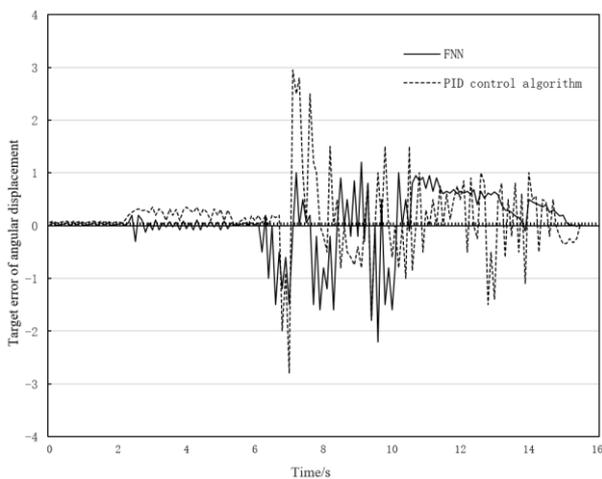


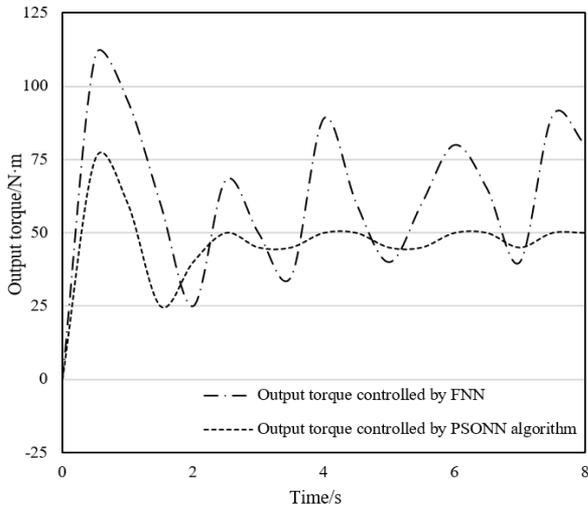
Figure 6. The comparison of the tracking errors of displacement at the end of two joints



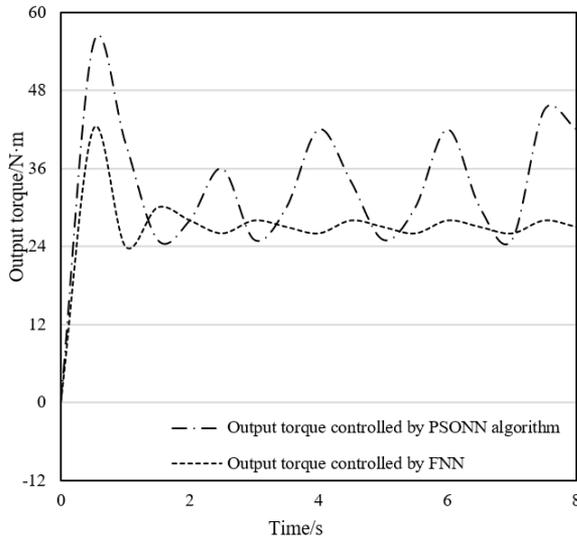
(a) The first joint

Figure 7 compares the output torques at the end of two joints under the control of our algorithm and the FNN, respectively. It can be seen that, before being optimized by adaptive PSO algorithm, the FNN control resulted in large fluctuations in the output torques of the first and second joints. After the optimization, the jitter phenomena were obviously mitigated.

Therefore, the trajectory of medical manipulator planned by our algorithm can suppress external disturbances as per the needs of the actual objective, improve the ability of jitter suppression, and enhance the positioning accuracy of each joint.

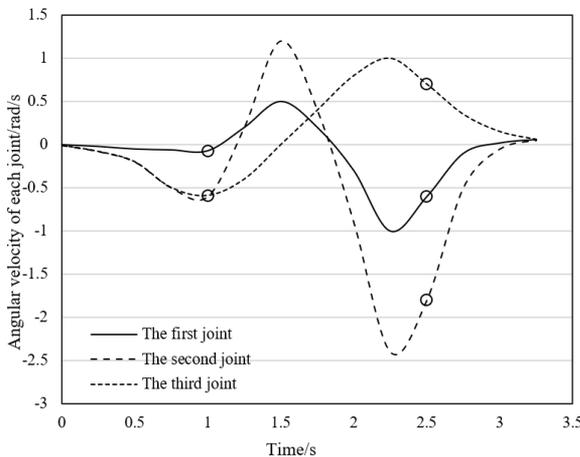


(a) The first joint



(b) The second joint

**Figure 7.** The comparison of the output torques at the end of two joints



**Figure 8.** The monitored angular velocities of three joints

Figure 8 presents the monitored angular velocities of the three joints of the manipulator. Obviously, the angular velocity of every joint stabilized after 2-3s, indicating that our algorithm can rapidly plan a suitable trajectory for the medical manipulator.

## 6. CONCLUSIONS

This paper designs a trajectory planning algorithm for medical manipulators by optimizing FNN with an adaptive PSO algorithm. Firstly, a kinetic model was constructed for an n-DOF medical manipulator based on position and posture transforms, and used to construct an FNN for trajectory planning. Then, an adaptive PSO algorithm was developed to suppress the jitter at the end of the medical manipulator, and used to optimize the FNN. The steps of the PSO algorithm were described in details. Finally, the performance of the proposed PSO algorithm was tested through experiments on a 3DOF medical manipulator. The experimental results show that our algorithm outperforms the traditional PID control algorithm in tracking error. Moreover, our algorithm was found to have a good ability to suppress the end jitter of the manipulator, enhance jitter suppression and joint positioning accuracy, and achieve a good effect of velocity control.

## ACKNOWLEDGMENT

This work was supported by the Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No.: KJQN201903108, KJ1602901), Chongqing College of Electronic Engineering Scientific Research Project (Grant No.: XJZK201809) and Xinxiang Medical University Education and Teaching Reform Research Project (Grant No.: 2017-XYJG-41).

## REFERENCES

- [1] Kamali, K., Joubair, A., Bonev, I. A., Bigras, P. (2016). Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 4320-4327. <https://doi.org/10.1109/ICRA.2016.7487630>
- [2] Faulwasser, T., Weber, T., Zometa, P., Findeisen, R. (2016). Implementation of nonlinear model predictive path-following control for an industrial robot. IEEE Transactions on Control Systems Technology, 25(4): 1505-1511. <https://doi.org/10.1109/TCST.2016.2601624>
- [3] Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A.M., Zanero, S. (2017). An experimental security analysis of an industrial robot controller. In 2017 IEEE Symposium on Security and Privacy (SP), pp. 268-286. <https://doi.org/10.1109/SP.2017.20>
- [4] Guillo, M., Dubourg, L. (2016). Impact & improvement of tool deviation in friction stir welding: Weld quality & real-time compensation on an industrial robot. Robotics and Computer-Integrated Manufacturing, 39: 22-31. <https://doi.org/10.1016/j.rcim.2015.11.001>

- [5] Yin, X., Pan, L. (2018). Enhancing trajectory tracking accuracy for industrial robot with robust adaptive control. *Robotics and Computer-Integrated Manufacturing*, 51: 97-102. <https://doi.org/10.1016/j.rcim.2017.11.007>
- [6] Su, Y.H., Chen, C.Y., Cheng, S.L., Ko, C.H., Young, K.Y. (2018). Development of a 3D AR-based interface for industrial robot manipulators. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1809-1814. <https://doi.org/10.1109/SMC.2018.00313>
- [7] Ayten, K.K., Dumlu, A. (2018). Real-time implementation of sliding mode control technique for two-DOF industrial robotic arm. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 8(4): 77-85.
- [8] Koumboulis, F.N. (2018). On the common control design of robotic manipulators carrying different loads. In International Conference on Robotics in Alpe-Adria Danube Region, pp. 416-424. [https://doi.org/10.1007/978-3-030-00232-9\\_44](https://doi.org/10.1007/978-3-030-00232-9_44)
- [9] Sefati, S., Alambeigi, F., Iordachita, I., Taylor, R.H., Armand, M. (2018). On the effect of vibration on shape sensing of continuum manipulators using fiber Bragg gratings. In 2018 International Symposium on Medical Robotics (ISMR), pp. 1-6. <https://doi.org/10.1109/ISMR.2018.8333303>
- [10] Wahrburg, A., Bös, J., Listmann, K. D., Dai, F., Matthias, B., Ding, H. (2017). Motor-current-based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control. *IEEE Transactions on Automation Science and Engineering*, 15(2): 879-886. <https://doi.org/10.1109/TASE.2017.2691136>
- [11] Capurso, M., Ardakani, M.M.G., Johansson, R., Robertsson, A., Rocco, P. (2017). Sensorless kinesthetic teaching of robotic manipulators assisted by observer-based force control. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 945-950. <https://doi.org/10.1109/ICRA.2017.7989115>
- [12] Yang, T., Yang, Z., Li, T., Liu, S., Wang, Y. (2018). Research on cooperation control of lightweight manipulator. In 2018 37th Chinese Control Conference (CCC), pp. 5498-5503. <https://doi.org/10.23919/ChiCC.2018.8482958>
- [13] Mareczek, J. (2019). *Grundlagen der roboter-manipulatoren-band 2: Pfad-und bahnplanung, antriebsauslegung, regelung*. Springer Berlin Heidelberg.
- [14] Bianco, C.G.L., Raineri, M. (2017). An automatic system for the avoidance of wrist singularities in anthropomorphic manipulators. In 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pp. 1302-1309. <https://doi.org/10.1109/COASE.2017.8256280>
- [15] Navarro, B., Cherubini, A., Fonte, A., Poisson, G., Fraise, P. (2017). A framework for intuitive collaboration with a mobile manipulator. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6293-6298. <https://doi.org/10.1109/IROS.2017.8206532>
- [16] Mohamed, Z., Khairudin, M., Husain, A.R., Subudhi, B. (2016). Linear matrix inequality-based robust proportional derivative control of a two-link flexible manipulator. *Journal of Vibration and Control*, 22(5): 1244-1256. <https://doi.org/10.1177%2F1077546314536427>
- [17] Annisa, J., Darus, I.M., Tokhi, M.O., Mohamaddan, S. (2018). Implementation of PID based controller tuned by evolutionary algorithm for double link flexible robotic manipulator. In 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), pp. 1-5. <https://doi.org/10.1109/ICASSDA.2018.8477615>
- [18] Jamali, A., Mat Darus, I.Z., Mohd Samin, P.P., Tokhi, M.O. (2018). Intelligent modeling of double link flexible robotic manipulator using artificial neural network. *Journal of Vibroengineering*, 20(2): 1021-1034. <https://doi.org/10.21595/jve.2017.18575>
- [19] Annisa, J., Abidin, A.Z., Darus, I.M., Tokhi, M.O. (2018). Controlling the non-parametric modeling of double link flexible robotic manipulator using hybrid PID tuned by P-Type ILA. *International Journal of Integrated Engineering*, 10(7): 219-232. <https://doi.org/10.30880/ijie.2018.10.07.020>
- [20] Mohammed, A.A., Eltayeb, A. (2018). Dynamics and control of a two-link manipulator using pid and sliding mode control. In 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), pp. 1-5. <https://doi.org/10.1109/ICCCEEE.2018.8515795>
- [21] Bagheri, M., Naseradinmousavi, P., Krstić, M. (2019). Feedback linearization based predictor for time delay control of a high-DOF robot manipulator. *Automatica*, 108: 108485. <https://doi.org/10.1016/j.automatica.2019.06.037>
- [22] Pizarro-Lerma, A.O., Garcia-Hernandez, R., Santibanez, V., Chin, J.V. (2019). Experimental evaluation of a sectorial fuzzy controller plus adaptive neural network compensation applied to a 2-DOF robot manipulator. *IFAC-PapersOnLine*, 52(29): 233-238. <https://doi.org/10.1016/j.ifacol.2019.12.655>
- [23] Campisano, F., Ramirez, A., Landewee, C.A., Caló, S., Obstein, K., Webster III, R.J., Valdastrì, P. (2020). Teleoperation and Contact Detection of a Waterjet-Actuated Soft Continuum Manipulator for Low-Cost Gastroscopy. *IEEE Robotics and Automation Letters*, 5(4): 6427-6434. <https://doi.org/10.1109/LRA.2020.3013900>
- [24] Wilkening, P., Alambeigi, F., Murphy, R.J., Taylor, R.H., Armand, M. (2017). Development and experimental evaluation of concurrent control of a robotic arm and continuum manipulator for osteolytic lesion treatment. *IEEE Robotics and Automation Letters*, 2(3): 1625-1631. <https://doi.org/10.1109/LRA.2017.2678543>
- [25] Sefati, S., Murphy, R.J., Alambeigi, F., Pozin, M., Iordachita, I., Taylor, R.H., Armand, M. (2018). FBG-based control of a continuum manipulator interacting with obstacles. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6477-6483. <https://doi.org/10.1109/IROS.2018.8594407>