IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 7, 2020, accepted July 11, 2020, date of publication July 15, 2020, date of current version July 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009526

An Online Trajectory Planning Method of a Flexible-Link Manipulator Aiming at Vibration Suppression

YUANYUAN LI[1], SHUZHI SAM GE[1,3], QINGPING WEI[2], TAO GAN[1], AND XIAOLIN TAO[1]

[1]Center for Robotics, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]DEC Academy of Science and Technology Company Ltd., Chengdu 611731, China
[3]Department of Electrical Computer Engineering, National University of Singapore, Singapore 119077

Corresponding author: Shuzhi Sam Ge (samge@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1813202 and Grant 61773093, in part by the National Key Research and Development Program of China under Grant 2018YFC0831800, in part by the Research Programs of Sichuan Science and Technology Department under Grant 17ZDYF3184, and in part by the Important Science and Technology Innovation Projects in Chengdu under Grant 2018-YF08-00039-GX.

ABSTRACT Vibration has a great influence on the working accuracy of flexible-link manipulators. In this paper, an online trajectory planning method for the flexible-link manipulator is proposed to suppress the vibration. Firstly, the vibration dynamic model of planar flexible-link manipulator is established, and the actual vibration trajectory of manipulator is solved by reasonable simplification. Then, taking residual energy of vibration as an objective function, the optimizer based on Particle Swarm Optimization(PSO) intelligent search algorithm is employed to search the motion trajectory of the manipulator with the best vibration suppression effect. Finally, the optimizer based on back propagation neural networks(BPNN) is employed instead of PSO optimizer to generate manipulator motion trajectory rapidly, and the optimal trajectory database searched by PSO optimizer is used to train BPNN. Simulation results show that this method can generate the trajectory rapidly and suppress the vibration effectively, and can be applied to the online trajectory planning of flexible manipulator.

INDEX TERMS Vibration suppression, online trajectory planning, flexible-link manipulator, particle swarm optimization, neural networks.

I. INTRODUCTION

Nowadays, with the diversified development of robots, the structural flexibility of robots can't be ignored. In social robots and cooperative robots, due to the consideration of safety in cooperative work, the self weight of robots is becoming lighter and lighter, and composite materials are being used in some structure, which makes the flexibility of robot links cannot be ignored [1], [2]. For the space manipulator, due to the weight requirements of the launch vehicle and space operation tasks, the self weight of the manipulator should be as light as possible, and length of it should be quite long, which increase the flexibility of the manipulator structure [3]–[8]. For the nuclear environment robot, in order to meet the needs of task and obstacle avoidance, the robot needs to control its own volume and increase working radius, which also makes the flexibility of the manipulator cannot be ignored [9], [10]. For the surgical robot, because of its small size and high requirement of operation precision, the flexibility of robot should also be considered [11], [12]. Due to the existence of structural flexibility, the manipulator will inevitably produce structural vibration in the process of motion, which will affect the accuracy of motion. Therefore, we need to consider the vibration effect duing the control process of the manipulator.

Vibration suppression of manipulator is an important research issue, and many research results have been produced in recent years. In references [13] and [14], adaptive control method is used to solve the control problem of flexible beam system or spring system with input and output boundary constraints. The open-loop vibration suppression method based on energy method is proposed in document [15] for multi link flexible manipulator system. For the series flexible joint manipulator, the vibration evaluation function is established based on the energy method in reference [16] and [17], and the optimal path is searched by particle swarm optimization algorithm. For the vibration suppression of a single flexible

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

link element, the vibration evaluation function is established based on the energy method in document [18], and the optimal path is searched by genetic algorithm. However, there are few papers on the open-loop vibration suppression of the flexible link manipulator with seies multiple degrees of freedom.

The dynamic modeling of flexible manipulator is a complex mathematical physics problem. In recent years, people have studied the dynamic modeling of flexible-joint and flexible-link manipulator. In document [19], the dynamic model of the robot's lower limb system with flexible joints is established, and the motor parameters of the joints are defined. In reference [20], the flexible intermediate link is modeled as Euler Bernoulli beam with pinning boundary conditions by using the assumed mode method, and the dynamic model of 3-PRR parallel manipulator with three flexible intermediate links is established. In reference [21], the dynamic model of space flexible link manipulator is established using Lagrangian method, and its vibration response is studied. In reference [22], a dynamic modeling method of flexible link manipulator based on artificial neural networks is proposed, and the neural identifier designed by this method can generate an approximate dynamic model by identifying the dynamic characteristics of the manipulator. However, the vibration evaluation method of the multi link flexible manipulator in series under any path has not been effectively solved.

Due to the complexity of robot dynamics model, intelligent optimization algorithm plays an effective role in solving the optimization problem of robot vibration control. Yue *et al.* [23] studies the problem of optimal trajectory planning of redundant robot with constraints based on genetic algorithm, and the joint trajectory is described by quartic and quintic polynomials. The coefficients of joint trajectory polynomials are determined by genetic algorithm, and the simulation verification is carried out on a plane three link flexible manipulator. In document [24], a new particle swarm optimization (PSO) algorithm is proposed to search for the optimal trajectory of flexible-joint manipulator to minimize the elastic potential energy involved in the motion. In document [25], the finite element difference method is used to establish the vibration dynamic model of piezo actuated cantilever beam, and the parameters of the fuzzy controll are optimized by particle swarm optimization algorithm, which achieves good vibration suppression effect.

Neural networks has the function of fitting any complex nonlinear mapping. In 1989, Robert Hecht-Nielsen proved that a continuous function in any closed interval can be approximated by BP networks which has a hidden layer. Now the neural networks has been widely used in the field of objective detection, robot parameter recognition and motion control. In references [26] and [27], tool dynamics of the surgicalrobot and human motions in dynamic situations are identified respectively by using neural networks. In the research of manipulator parameter identification, BP neural networks can identify the dynamic inertia parameters of manipulator intelligently [28]. In the research of manipulator kinematics control, trained BP neural networks can be used to solve the inverse kinematics of manipulator instead of the traditional analytical methods [29], [30]. When multi-robots work together, BP neural networks can assign tasks quickly according to the requirements of tasks [31]. In the master-slave control of robots, BP networks can be a predictor on the master side to predict the state of the slave side, which can reduce the influence of the information feedback delay during the remote feedback process [32]. At present, BP neural networks have not been used for vibration suppression effectively.

In this paper, the vibration suppression of a planar multi-DOF serial manipulator with flexible links is studied. Firstly, the dynamic model of a plane serial manipulator with flexible links is established, then the vibration evaluation function of the plane serial manipulator is established based on residual energy method. Then the off-line optimal trajectory search method is carried out by using a particle swarm optimization(PSO) algorithm. Finally, using the off-line optimal trajectory database as the training samples, an on-line vibration suppression path planning method based on back propagation neural networks(BPNN) is established. The simulation results verify the effectiveness of the vibration suppression method.

The major contributions of this paper are summarized as follows:

1. The vibration dynamic model of flexible-links planar serial manipulator is established, and the analytical solution is obtained on the premise of reasonable simplification, which makes it possible to evaluate the vibration during motion;

2. Based on BPNN, a rapid trajectory generator which can be used in on-line system is established, and an off-line training system for BPNN solver is established by PSO algorithm based on residual energy function.

II. PROBLEM FORMULATION

The mechanism diagram of a n-DOF flexible-link planar serial manipulator can be simplifed to Figure 1. Each arm is abstracted as a flexible link, and the centroid of each member is approximately located on the axis of the straight link. In Figure 1, O_i, m_i, and l_i respectively represent the coordinate origin, mass, length of the i-th link; θ_i represents the rotation angle relative of the i-th link to its previous link; l_{ig} reprents the distance from the mass center of the i-th link to its rotation hinge point.

Under the assumption of small elastic deformation, the first-order vibration mode of flexible link plays a leading role. Consider only the first-order vibration, link i can be simplified as the spring system shown in Figure 2. θ_i is the driving angle of the link i; φ_i is the actual angle under the flexibility of link; the torsion spring has a elastic coefficient k_i, and it is in the balanced position when θ_i and φ_i are equal. The elastic coefficient of flexible link can be easily obtained by elastic mechanics theory or experiment. The vibration dynamic equation of link i can be expressed as equation 1.

$$T_i = -k_i * (\theta_i - \theta_{id}) \tag{1}$$

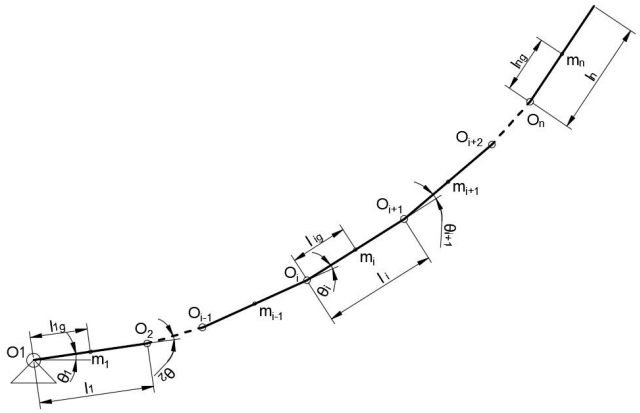


FIGURE 1. Vibration trajectory of manipulator end under QC method.

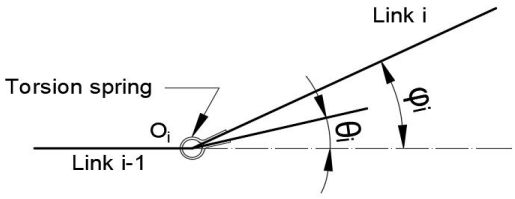


FIGURE 2. Simplified model of link i.

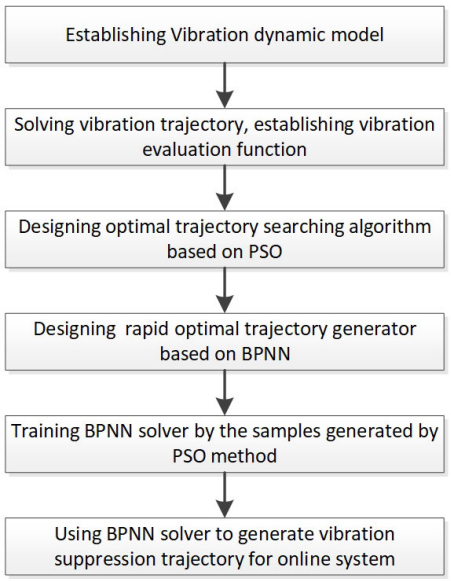


FIGURE 3. Algorithm framework.

III. CONTROL DESIGN

In this section, we establish an on-line vibration suppression trajectory planning method based on particle swarm optimization(PSO) and shallow back-propagation neural networks(BPNN). Firstly, the vibration dynamic equation of the planar series manipulator is established, and the approximate vibration trajectory under any input trajectory is solved. Then, an residual energy method is used to evaluate the path vibration of the planar series manipulator. Based on PSO search algorithm, the best interpolation points are searched for the known starting and ending conditions of motion, and the optimal input trajectory of vibration suppression is obtained by smooth connection of interpolation points. Finally, the vibration suppression algorithm based on neural networks is established for the intelligent generation of interpolation points. The neural networks is trained by the sample database established by PSO search algorithm. The control design process is shown as Figure 3.

A. VIBRATION EVALUATION FOR MANIPULATOR MOTION

In this section, firstly, the dynamic model of the planar series manipulator is established, and the actual vibration trajectory of the manipulator under input motion trajectory is calculated. Then the vibration evaluation function of input trajectory is established based on the energy method.

Thereom 1: For the n-DOF flexible planar tandem manipulator shown as Figure 4, only the first-order vibration of each link is considered, and the interaction between the vibration of each link is ignored, then the vibration dynamic equation

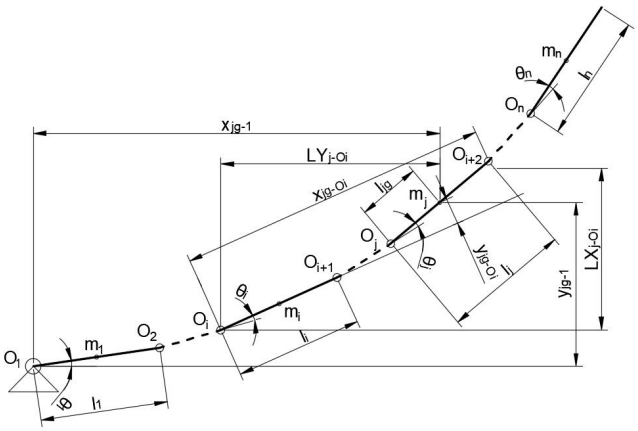


FIGURE 4. Manipulator model.

of any joint i can be expressed as equation 2.

$$\ddot{\varphi_i}(t) * \sum_{j=1}^{n} I_{joi} + k_i * \varphi_i(t) = -\sum_{j=1}^{n} m_j * \left(f_1 * LY_{j-oi} + f_2 * LX_{j-oi}\right) - \sum_{j=i+1}^{n} m_j * f_3 * x_{jg-oi} + k_i * \theta_i(t) \quad (2)$$

where,

$$f_1 = \sum_{k=1}^{j-1} \left(\frac{\partial c_1}{\partial \theta_k} * \frac{d\theta_k}{dt} + \frac{\partial c_1}{\partial \dot{\theta_k}} * \frac{d\dot{\theta_k}}{dt} \right) \quad (3)$$

$$f_2 = \sum_{k=1}^{j-1} \left(\frac{\partial c_2}{\partial \theta_k} * \frac{d\theta_k}{dt} + \frac{\partial c_2}{\partial \dot{\theta_k}} * \frac{d\dot{\theta_k}}{dt} \right) \quad (4)$$

$$f_3 = \sum_{k=i+1}^{j} \left(\frac{\partial c_3}{\partial \theta_k} * \frac{d\theta_k}{dt} + \frac{\partial c_3}{\partial \dot{\theta_k}} * \frac{d\dot{\theta_k}}{dt} \right) \quad (5)$$

$$c_1 = \sum_{k=1}^{j-1} \left(\frac{\partial y_{jg-1}}{\partial \theta_k} * \frac{d\theta_k}{dt} \right) \quad (6)$$

$$c_2 = \sum_{k=1}^{j-1}\left(\frac{\partial x_{jg-1}}{\partial \theta_k} * \frac{\mathrm{d}\theta_k}{\mathrm{d}t}\right) \tag{7}$$

$$c_3 = \sum_{k=i+1}^{j}\left(\frac{\partial y_{jg-Oi}}{\partial \theta_k} * \frac{\mathrm{d}\theta_k}{\mathrm{d}t}\right) \tag{8}$$

$$I_{jOi} = \begin{cases} m_i * l_{ig}^2, & j = i \\ m_j * d_{Oi-gj}^2, & j = i+1 \sim n \end{cases} \tag{9}$$

$$LY_{j-oi} = \sum_{k=i}^{j-1}\left(l_k * cos\left(\sum_{i=1}^{k}\theta_i\right)\right) + l_{jg} * cos\left(\sum_{i=1}^{j}\theta_j\right) \tag{10}$$

$$LX_{j-oi} = \sum_{k=i}^{j-1}\left(l_k * sin\left(\sum_{i=1}^{k}\theta_i\right)\right) + l_{jg} * sin\left(\sum_{i=1}^{j}\theta_j\right) \tag{11}$$

$$d_{Oi-gj} = \left(d_{ij}^2 + l_{jg}^2 + 2 * d_{ij} * l_{jg} * cos\theta_{ji}\right)^{0.5} \tag{12}$$

$$d_{ij} = \begin{cases} I_i, & j = i+1 \\ \left(d_{i(j-1)}^2 + l_{j-1}^2 + 2 * d_{i(j-1)} * \right. & \\ \left. l_{j-1} * \cos\theta_{(j-1)i}\right)^{0.5}, & j = i+2 \sim n \end{cases} \tag{13}$$

$$\cos\theta_{ji} = \begin{cases} \theta_i, j = i+1 \\ \theta_j + \dfrac{d_{i(j-1)}}{d_{ij}} * \left(\pi - \theta_{(j-1)i}\right), & j = i+2 \sim n \end{cases} \tag{14}$$

φ_i is the actual rotation angle of link i; θ_i is the driving rotation angle of link i.(φ_i and θ_i are not same because of the flexibility of the link i.) k_i is the elastic coefficient of link i; m_j is the mass of link i. $x_{jg-1}, y_{jg-1}, x_{jg-oi}$ and x_{jg-oi} is shown as Figure 4.

Proof: The torque of joint i is caused by the trajectory of the centroid of joint $i \sim n$.

$$T_i(t) = \sum_{j=i}^{n} T_{ij} \tag{15}$$

where, T_{ij} is the torque of joint i caused by the trajectory of joint j.

According to the motion-superposition principle, the motion of link j centroid could be divided into three parts: $motion_{jb}$, $motion_{js}$ and $motion_{ja}$, which respectively represent the motion of link j centroid caused by motion of link $1 \sim i-1$, link i, and link $i+1 \sim j$. Let T_{ijb}, T_{ijs} and T_{ija} respectively represent the torque of link i caused by $motion_{jb}$, $motion_{js}$ and $motion_{ja}$, then we can get

$$\begin{aligned} T_i(t) &= \sum_{j=i}^{n} T_{ij} = \sum_{j=i}^{n}(T_{ijb}) + \sum_{j=i}^{n}(T_{ijs}) + \sum_{j=i}^{n}(T_{ija}) \\ &= T_{ib} + T_{is} + T_{ia} \end{aligned} \tag{16}$$

where, T_{ib}, T_{is}, and T_{ia} represent the torque of link i caused by the motion of joint $1 \sim i-1$, joint i, and joint $i+1 \sim j$, respectively.

Obviously, when $i = 1$, $T_{ib} = 0$; when $i = n$, $T_{ia} = 0$.

For T_{ib}, take node O as the origin (coincide with node O_1) and the horizontal line as the X axis to establish the coordinate system, named C_1(shown as Figure 5).

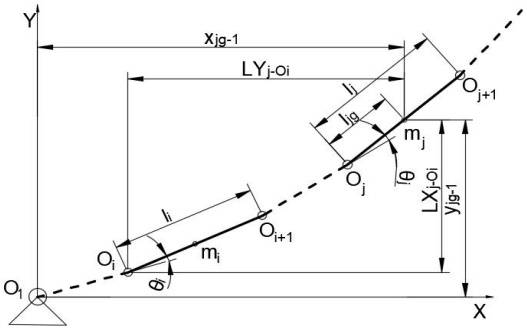


FIGURE 5. Coordinate C1.

According to the theory of rigid body dynamics, torque T_{ib} can be supressed as equation 17.

$$T_{ib}(t) = \sum_{j=i}^{n} m_j * \left(\ddot{y_{jg}}_{-1} * LY_{j-Oi} + \ddot{x_{jg}}_{-1} * LX_{j-Oi}\right) \tag{17}$$

where, y_{jg-1} is Y-axis value of the centroid of the link j in coordinate system C_1; x_{jg-1} is X-axis value of the centroid of link j in coordinate system C_1; LY_{j-Oi} is the force arm from Y direction at the centroid of link j to point O_i, as shown in Equation 10; LX_{j-Oi} is the force arm from X direction at the centroid of link j to point Oi, as shown in Equation 11.

The values of y_{jg-1} and x_{jg-1} are calculated as follows.

$$y_{jg-1} = \sum_{k=1}^{j-1}\left(l_k * sin\left(\sum_{i=1}^{k}\theta_i\right)\right) + l_{jg} * sin\left(\sum_{k=1}^{j}\theta_k\right) \tag{18}$$

$$x_{jg-1} = \sum_{k=1}^{j-1}\left(l_k * cos\left(\sum_{i=1}^{k}\theta_i\right)\right) + l_{jg} * cos\left(\sum_{k=1}^{j}\theta_k\right) \tag{19}$$

Tib is the torque that produced by link $i \sim i-1$, so the derivation of acceleration in this function is only for $\theta_1 \sim \theta_{i-1}$. The torque generated by the rotation of other links will be considered in the subsequent part.

$$\ddot{y_{jg}}_{-1} = \sum_{k=1}^{j-1}\left(\frac{\partial \dot{y_{jg}}_{-1}}{\partial \theta_k} * \frac{\mathrm{d}\theta_k}{\mathrm{d}t} + \frac{\partial \dot{y_{jg}}_{-1}}{\partial \dot{\theta_k}} * \frac{\mathrm{d}\dot{\theta_k}}{\mathrm{d}t}\right) \tag{20}$$

$$\ddot{x_{jg}}_{-1} = \sum_{k=1}^{j-1}\left(\frac{\partial \dot{x_{jg}}_{-1}}{\partial \theta_k} * \frac{\mathrm{d}\theta_k}{\mathrm{d}t} + \frac{\partial \dot{x_{jg}}_{-1}}{\partial \dot{\theta_k}} * \frac{\mathrm{d}\dot{\theta_k}}{\mathrm{d}t}\right) \tag{21}$$

For T_{is}, the motion of link $i \sim n$ caused by link i is a single rotation motion around node O_i. Take node O_i as the origin and the direction $O_{i-1} - O_i$ as the X axis to establish the coordinate system, named C_2(shown as Figure 6).

According to the theory of rotational dynamics, torque T_{is} can be supressed as below.

$$T_{is} = \ddot{\theta_i} * \sum_{j=i}^{n} I_{jOi} = \ddot{\varphi_2} * fis(t) \tag{22}$$

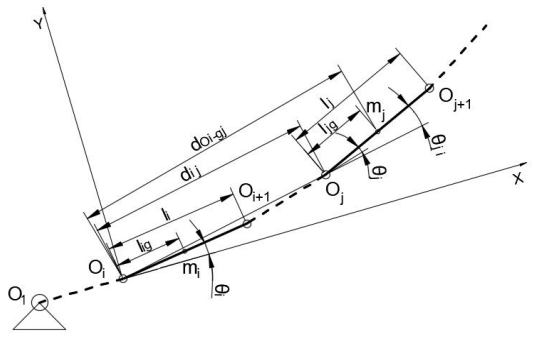


FIGURE 6. Coordinate C2.

where, I_{joi} is the rotational inertia of the rod j to point O_i, as shown in Equation 9.

For Tia, take node O_i as the origin and the direction $O_i - O_{i+1}$ as the X axis to establish the coordinate system, named C_3(shown as Figure 7).

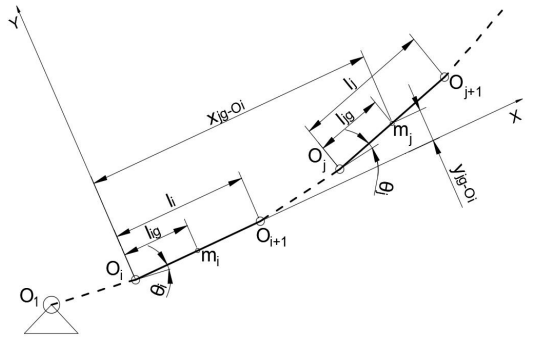


FIGURE 7. Coordinate C3.

According to the theory of rigid body dynamics, torque T_{is} can be supressed is shown below.

$$T_{ia} = \sum_{j=i+1}^{n} m_j * \ddot{y}_{jg-Oi} * x_{jg-Oi} \tag{23}$$

where, y_{jg-Oi} is Y-axis value of the centroid of link j in coordinate C_3; x_{jg-Oi} is X-axis value of the centroid of link j in coordinate C_3.

The values of x_{jg-Oi} and y_{jg-Oi} are calculated as follows.

$$x_{jg-Oi} = l_i + \sum_{k=i+1}^{j-1} \left(l_k * \cos\left(\sum_{l=i+1}^{k} \theta_l \right) \right) + l_{jg} * \cos\left(\sum_{l=i+1}^{j} \theta_l \right) \tag{24}$$

$$y_{jg-Oi} = \sum_{k=i+1}^{j-1} \left(l_k * \sin\left(\sum_{l=i+1}^{k} \theta_l \right) \right) + l_{jg} * \sin\left(\sum_{l=i+1}^{j} \theta_l \right) \tag{25}$$

$\ddot{y}_{jg-Oi}$ is the second derivative of y_{jg-Oi} to time t, means Y-axis acceleration of the centroid of link j in coordinate C_3.

$$\ddot{y}_{jg-Oi} = \sum_{k=i+1}^{j} \left(\frac{\partial \ddot{y}_{jg-Oi}}{\partial \theta_k} * \frac{\mathrm{d}\theta_k}{\mathrm{d}t} + \frac{\partial \ddot{y}_{jg-Oi}}{\partial \dot{\theta}_k} * \frac{\mathrm{d}\dot{\theta}_k}{\mathrm{d}t} \right) \tag{26}$$

The torque of link i is the sum of T_{ib}, T_{is} and T_{ia}.

$$\begin{aligned} T_i &= T_{ib} + T_{is} + T_{ia} \\ &= \sum_{j=1}^{n} m_j * \left(\ddot{y}_{jg-i} * LY_{j-oi} + \ddot{x}_{jg-i} * LX_{j-oi} \right) \\ &\quad + \ddot{\varphi}_i * \sum_{j=i}^{n} I_{jOi} + \sum_{j=i+1}^{n} m_j * \ddot{y}_{jg-oi} * x_{jg-oi} \end{aligned} \tag{27}$$

Only the first-order vibration of the manipulator is considered and the mechanical damping effect is ignored, then the link i can be regarded as the first-order torsion spring system, as shown in Figure 4. Spring torque of link i is calculated as equation 28.

$$T_i = -k_i \left(\varphi_i - \theta_i \left(t \right) \right) \tag{28}$$

By combining equation 27 and 28, and reorganize them, equation 2 is obtained.

Theorem 2: For the flexible planar series manipulator with the form shown in Figure 1, divide the whole period of motion into m equal parts, which make the time interval of each period δt is tiny enough, and the actual vibration trajectory of any joint i can be approximated as below.

$$\varphi_i \left(t \right) = C_1 \cos\left(\sqrt{A} * t \right) + C_2 \sin\left(\sqrt{A} * t \right) + x^* \tag{29}$$

where

$$\begin{aligned} x^* &= \frac{a_5}{k_i} t^5 + \frac{a_4}{k_i} t^4 + \frac{1}{k_i} \left(a_3 - \frac{20}{A} a_5 \right) t^3 \\ &\quad + \frac{1}{k_i} \left(a_2 - \frac{12}{A} a_4 \right) t^2 + \frac{1}{ki} \left(a_1 - \frac{6}{A} a_3 + \frac{120}{A^2} a^5 \right) t \\ &\quad + \frac{1}{ki} \left(a_0 - \frac{2}{A} a_2 + \frac{24}{A^2} a_4 \right) \end{aligned} \tag{30}$$

$$A = \frac{k_i}{fis\left(t_j \right)} \tag{31}$$

$$fis\left(t \right) = \sum_{j=i}^{n} I_{jOi} \tag{32}$$

$$\begin{aligned} a_0 &= -\frac{fir^{(5)}\left(t_j \right) * t_j^5}{120} + \frac{fir^{(4)}\left(t_j \right) * t_j^4}{24} - \frac{\dddot{fir}\left(t_j \right) * t_j^3}{6} \\ &\quad + \frac{\ddot{fir}\left(t_j \right) * t_j^2}{2} - \dot{fir}\left(t_j \right) * t_j + fir\left(t_j \right) \end{aligned} \tag{33}$$

$$\begin{aligned} a_1 &= \frac{fir^{(5)}\left(t_j \right) * t_j^4}{24} - \frac{fir^{(4)}\left(t_j \right) * t_j^3}{6} + \frac{\dddot{fir}\left(t_j \right) * t_j^2}{2} \\ &\quad - \ddot{fir}\left(t_j \right) * t_j + \dot{fir}\left(t_j \right) \end{aligned} \tag{34}$$

$$a_2 = -\frac{fir^{(5)}(t_j) * t_j^3}{12} + \frac{fir^{(4)}(t_j) * t_j^2}{4} - \frac{\dddot{fir}(t_j) * t_j}{2} + \frac{\ddot{fir}(t_j)}{2} \quad (35)$$

$$a_3 = \frac{fir^{(5)}(t_j) * t_j^2}{12} - \frac{fir^{(4)}(t_j) * t_j}{6} - \frac{\dddot{fir}(t_j)}{6} \quad (36)$$

$$a_4 = -\frac{fir^{(5)}(t_j) * t_j}{24} + \frac{fir^{(4)}(t_j)}{24} \quad (37)$$

$$a_5 = \frac{fir^{(5)}(t_j)}{120} \quad (38)$$

$$fir(t) = -\sum_{j=i}^{n} \left(m_j * \left(f_1 * LY_{jg-Oi} + f_2 * LX_{jg-Oi}\right)\right) - \sum_{j=i+1}^{n} \left(m_j * f_3 * x_{jg-Oi}\right) + k_i * \theta_i(t) \quad (39)$$

where, t_j is the starting time of the j-th interval which the time variable t of the equation is located. C_1 and C_2 are normally bright, which are determined by the starting states of the motion process.

Proof: The vibration equation of the flexible planar series manipulator shown in Figure is a second-order variable coefficient non-homogeneous differential equation, and the coefficient and non-homogeneous term are more complex, so the analytical formula can not be solved.Therefore, in order to facilitate the use of the project, it is necessary to simplify it properly.

In order to express conveniently, let the second-order coefficients of the vibration dynamic equation be expressed as equation 32, and let the nonhomogeneous terms be expressed as equation 39.

In order to reduce the complexity of the equation, the whole time period $T = t_f - t_0$ can be divided into tiny m equal parts, and the time interval of each period is δt. Considered that $fis(t)$ is constant in the range of δt.

Actually, $fis(t)$ is the moment of inertia of each joint around the O_i axis. When δt is small enough, there is little change in δt period, so this simplification is reasonable. Here, we express the second-order coefficient of the jth time period $[t_j, t_{j+1})$ as $fis(t_j)$.

In addition, the nonhomogeneous term $fir(t)$ is replaced by its Taylor expansion formula $fi(t)$ in the time period $[t_j, t_{j+1})$. In fact, when δt is small enough, the error between the Taylor expansion and the original function can be ignored, and the simplification is reasonable.

Here, we employ the fifth-order Taylor formula to replace the nonhomogeneous term. Fifth-order Taylor expansion of the jth time period $[t_j, t_{j+1})$ is shown as equation 40.

$$fir(t) \approx fir(t_j) + \dot{fir}(t_j) * (\text{t-tj}) + \frac{\ddot{fir}(t_j)}{2!} * (\text{t-tj})^2 + \frac{\dddot{fir}(t_j)}{3!} * (\text{t-tj})^3 + \frac{fir^{(4)}(t_j)}{4!} * (\text{t-tj})^4 + \frac{fir^{(5)}(t_j)}{5!} * (t-tj)^5 \quad (40)$$

By sorting out equation 40, formula 41 is obtained.

$$fir(t) \approx a_0 + a_1 t^2 + a_2 t^2 + a_3 t^3 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (41)$$

where, $a_0 \sim a_5$ is shown as equation 33–38.

Thus, in the j-th time period $[t_j, t_{j+1})$, the vibration dynamic model of the manipulator can be simplified as

$$\ddot{\varphi}_i * fistj + ki * \varphi_i = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \quad i = 1 \sim n,\ t = t_0 \sim t_f \quad (42)$$

In equation 42, the vibration dynamic model of the manipulator is simplified to the second order non-homogeneous linear differential equation with constant coefficient, and the non-homogeneous term is a polynomial of degree 5. The solution of the differential equation can be easily obtained as equation 29.

The vibration of flexible body can be described by the residual energy of the system. The residual energy of vibration can be defined as the energy other than kinetic energy generated by input motion, including vibrational kinetic energy and elastic potential energy.

The residual energy of the system at any time is equal to the sum of the residual potential energy and the residual kinetic energy.

Residual potential energy at any time is shown as follows.

$$V_1(t) = \frac{1}{2}\delta_t^T K \delta^t \quad (43)$$

Residual kinetic energy at any time is shown as follows.

$$V_2(t) = \frac{1}{2}\dot{\delta_t}^T M \dot{\delta_t} \quad (44)$$

where, $\delta = \left[\delta_1(t) \cdots \delta_n(t)\right]$ is the error vector of manipulator; $K = \left[K_1 \cdots K_n\right]$ is the elasticity coefficient vector of manipulator; $M = \left[M_1(t) \cdots M_n(t)\right]$ is the moment of inertia vector of manipulator at time t;$\delta_i(t) = x_i(t) - q_i(t)$, $i = 1\ n$ is rotation error of the ith joint at time t; $x_i(t)$ and $q_i(t)$ is the actual trajectory and input trajectory of the ith joint at time t respectively.

The residual energy of system at any time is shown as equation 45.

$$V(t) = V_1 t + V_2(t) \quad (45)$$

B. VIBRATION OPTIMIZATION SEARCHING ALGOSITHM BASED ON PSO

The optimal trajectory search of a multi-axis manipulator based on vibration suppression is a complex multi-dimensional global search problem. PSO algorithm is a kind of swarm intelligence algorithm, which can effectively solve most of the global optimization problems, and its operation process has the characteristics of parallelism, which is convenient for multithreaded servers to speed up the operation of a large number of data. Therefore, based on the principle of minimum residual vibration energy, a PSO is employed to search the optimal motion path of the manipulator.

1) OPTIMIZATION ALGORITHM FRAMEWORK AND PROCESS

Search algorithm framework consists of an initialization layer, a searching layer and an output layer. According to the state parameters of the start and end points, the initialization layer finds a group of relatively good interpolation points data by the method of quintic polynomial interpolation, and generates the random initial position of particle swarm based on the data; the search layer keeps updating the particle position and recording the optimal particle through the PSO algorithm based on the principle of the minimum value of the objective function; after the number of iterations is reached, the output Layer takes the last recorded optimal particle as the output data.

The framework of vibration optimization searching algorithm based on PSO is shown as Figure 8.

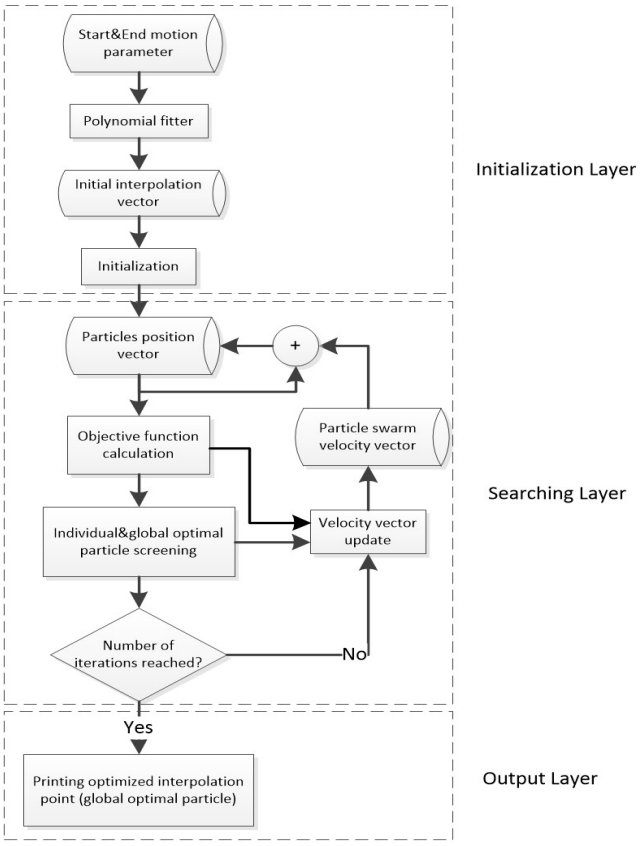


FIGURE 8. Framework of PSO algorithm.

2) INPUT CONDITIONS

The input conditions of the algorithm are the intrinsic parameters and the motion parameters of the system. The intrinsic parameters of the system are the parameters of the system itself, which do not change with the motion, such as the degree of freedom of the system, the length of the connecting rod, the moment of inertia, etc.; The motion parameters of the algorithm input are the position θ, velocity V, acceleration a and time t of the starting point and the ending point of each degree, which are expressed as equation III-B2.

$$Qi = \left[\theta_{i0}\ \omega_{i0}\ \alpha_{i0}\ t_{i0}\ \theta_{if}\ \omega_{if}\ \alpha_{if}\ t_{if}\right] \tag{46}$$

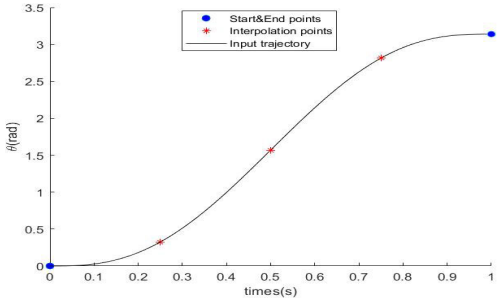


FIGURE 9. Input trajectory by quintic interpolation.

3) SEARCH VARIABLES

N interpolation points are set between the start point and the end point, and the parameters (position θ, angular velocity ω, angular acceleration α and time t) of the interpolation points are used as search variables. The interpolation points are connected in sequence as the input trajectory of the manipulator. In order to ensure the smoothness of the curve and the continuity of the position, velocity and acceleration of each interpolation point, the interpolation points are connected by a quintic polynomial function(shown in the figure 9). The form of quintic polynomial function connecting interpolation points is shown as equation 47.

$$\theta = a_5 * t^5 + a_4 * t^4 + a_3 * t^3 + a_2 * t^2 + a_1 * t + a_0 \tag{47}$$

The search variables can be expressed as equation 48.

$$X = \begin{bmatrix} \theta_{11} & \omega_{11} & \alpha_{11} & \cdots & \theta_{ij} & \omega_{ij} & \alpha_{ij} \\ \cdots & \theta_{mn} & \omega_{mn} & \alpha_{mn} &] & & \end{bmatrix} \tag{48}$$

4) INITIALIZATION OF SEARCH VARIABLES

In order to obtain a relatively good initial search value to improve the search effect, the initial trajectory is set to be a quintic polynomial curve, which keep the motion parameters of the start and end points be continuous and the motion trajectory be smooth. So, the start and end points of the trajectory are connected by a quintic polynomial function, and then the calculated quintic polynomial values ($f(tj)$,$\dot{f}(tj)$,$\ddot{f}(tj)$) at each interpolation time are took as the parameter values (position θ_j, angular velocity ω_j, angular acceleration α_j) of the interpolation point.

Let the initial trajectory equation be as shown in equation 49, where $a0 \sim a5$ is the coefficient of each order of the quintic polynomial. The values of $a0 \sim a5$ is obtained by solving equations 50.

$$\begin{aligned} \theta_i(t) = {} & a_{i5} * t^5 + a_{i4} * t^4 + a_{i3} * t^3 \\ & + a_{i2} * t^2 + a_{i1} * t + a_{i0} \end{aligned} \tag{49}$$

$$\begin{cases} \theta_i(t_0) = \theta_{i0} \\ \theta_i'(t_0) = \omega_{i0} \\ \theta_i''(t_0) = \alpha_{i0} \\ \theta_i(t_f) = \theta_{if} \\ \theta_i'(t_f) = \omega_{if} \\ \theta_i''(t_f) = \alpha_{if} \end{cases} \tag{50}$$

Then we can get the initial search variables vector X_0 shown as equation 51. The values of variables can be obtained by equation 52.

$$X_0 = \begin{bmatrix} \theta_{110} & \omega_{110} & \alpha_{110} & \cdots & \theta_{ij0} & \omega_{ij0} \\ \alpha_{ij0} & \cdots & \theta_{mn0} & \omega_{mn0} & \alpha_{mn0} & \end{bmatrix} \quad (51)$$

$$\begin{cases} \theta_{ij0} = \theta_i(t_j) \\ \omega_{ij0} = \theta_i'(t_j) \\ \alpha_{ij0} = \theta_i''(t_j) \end{cases} \quad (52)$$

5) OBJECTIVE FUNCTION

The residual energy is the root reason of the vibration caused by the motion of manipulator. In this scheme, the residual energy generated during the motion of the manipulator is taken as the objective function of the optimization algorithm. The objective function is shown as equation 53.

$$functionV = V_{tf} + \frac{1}{t_f - t_0} \int_{t_0}^{t_f} V_t \mathrm{d}t \quad (53)$$

It means the sum of the residual energy of the system at the termination time t_f and the average residual energy in the motion process. This objective function takes the vibration of the system when it stops as the main investigation factor, and takes into account vibration during the process of motion, which can well reflect the overall vibration level of the system in motion.

6) SEARCHING ALGORITHM

In the D-dimensional target search space, N particles form a community, and each particle i contains a D-dimensional position vector $x_k = \begin{bmatrix} x_{k1} & \cdots & x_{kD} \end{bmatrix}$ and velocity vector $v_k = \begin{bmatrix} v_{k1} & \cdots & v_{kD} \end{bmatrix}$. When searching the D-dimensional solution space, particle i saves the optimal position it finds. In each iteration, particle i adjusts its own velocity vector according to its own inertia, own experience and group optimal experience, so as to adjust its own position. We can use a fitness function to measure the superiority of particles. In this research, the dimention of position vector is $D = 3 * m * n$, where m is the number of degrees of freedom of robot and n is the number of interpolation points of joint trajectory.

$$x_k = \begin{bmatrix} \theta_{11k} & \omega_{11k} & \alpha_{11k} & \cdots & \theta_{ijk} & \omega_{ijk} \\ \alpha_{ijk} & \cdots & \theta_{mnk} & \omega_{mnk} & \alpha_{mnk} & \end{bmatrix} \quad (54)$$

where, θ_{ijk}, ω_{ijk} and α_{ijk} is the angle, angular velocity and angular acceleration of the j-th interpolation point of the i-th link in the k-th particle.

The PSO algorithm uses equation 55 to update the particles:

$$\begin{cases} v_{kd} = \omega v_i k + c_1 r_1 \left(p_{kd} - x_{kd}\right) + c_2 r_2 \left(p_{gd} - x_{kd}\right) \\ x_{kd} = x_{kd} + v_k d \end{cases} \quad k = 1 \sim p; d = 1 \sim D \quad (55)$$

In equation 55, c_1 and c_2 are the learning factors, whose value range is $(0, 1)$; r_1 and r_2 are random numbers, whose value range is $(0, 1)$; ω is the inertia weight, which is used to control the influence of the speed before the update on the speed after the update; $v_k d \in [-v_{max}, v_{max}]$, where $vmax$ is a constant, which is set according to the actual problem. The termination condition of PSO iteration is to reach the maximum number of PSO iterations set by the designer.

C. VIBRATION SUPPRESSION ALGORITHM BASED ON BPNN

Although the optimal path search algorithm based on a PSO algorithm can get excellent results, the computation amount is large and the search time is very long, which is due to the complexity of vibration dynamics equation, as well as the number of samples and iterations of the search algorithm. According to the test, using MATLAB software in the case of parallel operation of 24 thread CPU, the time to complete a trajectory search algorithm is 6-8 hours. This is unacceptable in engineering applications. In order to search fast online in the process of engineering application, we employ a shallow BPNN as the optimal path generator, and train it using the sample data of PSO algorithm.

1) ALGORITHM FRAMEWORK

The vibration suppression algorithm based on a BPNN consist of an initial vector Q, an interpolation vector X, an input vector IN, an input normalization vector INN, a BPNN solver, an adjustment normalization vector ADN, an adjustment vector AD, an output vector OU, and a QP interpolator. The framework of this algorithm is shown as Figure 10.

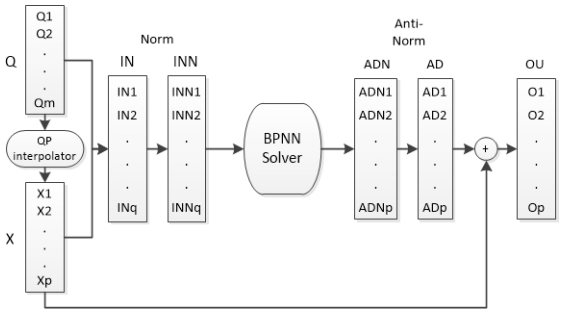


FIGURE 10. Framework of algorithm based on BPNN.

The initial vector Q is a vector of $8 * m$ dimension, whose values respectively represent the angle θ, angular velocity ω, angular acceleration α and time t of the start and end points of each degree of the manipulator. The composition of vector Q is as follows.

$$Q = \begin{bmatrix} Q_1 & \cdots & Q_i & \cdots & Q_m \end{bmatrix} \quad (56)$$

where, $Qi(i = 1 \sim m)$ is shown as equation III-B2.

In order to reduce the fitting difficulty of the neural networks, we further expand the input data, add the data of quintic polynomial interpolation points shown in equation 51 to the initial vector. The function of QP interpolator is to get the initial interpolation vector X_0 which is shown as equation 51,

and its calculation method has been detailed when we get the equation 51. By combining the vectors Q and X_0, we get the input vector IN which has $q = (8 + 3 * n) * m$ dimension.

$$IN = \left[Q \; X_0 \right] \tag{57}$$

The adjustment vector AD is a vector of $3*m*n$ dimension, whose value represents the value to be adjusted based on the interpolation vector X_0.

The output vector OU is a vector of $p = 3 * m * n$ dimension, whose value represents the interpolation vector with good vibration suppression effect after optimization by the algorithm. The output vector is obtained by adding the interpolation vector X_0 and the adjustment vector AD.

The BPNN solver is a solver based on back propagation neural networks, whose function is to solve the output layer according to the input layer and the known conditions of the system, so as to get the adjustment vector AD with good vibration suppression effect.

In order to achieve better learning and solving effect of BPNN solver, the input expansion data and adjustment data in the training samples of neural networks are normalized. Therefore, in the program framework, the input expansion vector IN needs to be normalized to get the input layer(vector INN). Similarly, the output layer(vector ADN) of the neural networks should also be de anti-normalized to get the adjustment vector AD.

2) BPNN SOLVER

The BPNN solver is shallow back-propagation neural networks, which consists of an input layer, a hidden layer and an output layer. The input layer is the normalized form of input expansion vector INN, the output layer is the normalized form of adjustment vector ADN, the hidden layer is composed of many nodes, and the number of nodes is based on experience and the matrix number of the input layer and the output layer.

The structure of the BPNN solver is shown as Figure 11.

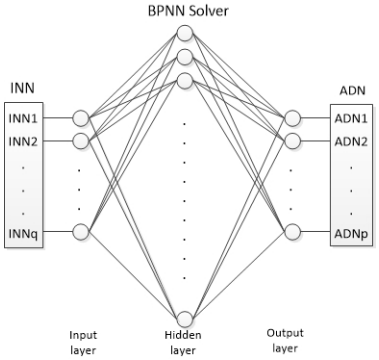


FIGURE 11. Structure of BPNN solver.

The mathematical expression of BPNN solver is shown as equation 58.

$$\begin{cases} s_j = \sum_{i=1}^{q} W_{ij}^1 * INN_i, & j = 1 \sim n_{hi} \\ ADN_p = \sum_{j=1}^{h_{hi}} W_{jk}^2 * f\left(s_j\right), & k = 1 \sim p \end{cases} \tag{58}$$

where, $s = [s_1, \dot{;} s_{n_h i}]$ is the hidden layer vector; $W_i^1 j$ is the coefficient from the i-th node of input layer to the j-th node of hidden layer; $W_j^2 k$ is the coefficient from the j-th node of hidden layer to the k-th node of output layer; $f(s)$ is the activation function of hidden layer.

The vibration dynamic model of the manipulator is a complex nonlinear mathematical function, so a nonlinear activation function should be employed to solve the optimal vibration path. The tanh function is employed as the activation function, as shown below.

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{59}$$

3) DATA NORMALIZATION

Only in the range of $[-1.7, 1.7]$ can the Sinh activation function have good nonlinearity, and beyond this range, the gradient of the activation function gradually disappears. However, the training sample data and the actual input data range of neural networks are not the same. If only the weights and position variables are used to adjust, the weights and position variables of each node will have a large order of magnitude difference. On the one hand, it will make the activation function of neural networks difficult to work in a good nonlinear range, on the other hand, it will make the step length in the learning process difficult adapt to each function, resulting in poor learning effect.

Therefore, we process the input and output data to control it between $[-1,1]$ to get better fitting effect. The data processing function is as equation 60.

$$x^* = \frac{x - x_{me}}{x_{ma} - x_{me}} \tag{60}$$

where, x^* is processed data; x is raw data; x_{av} is mean value; x_{ma} is maximum value.

When the result is calculated, the output of the neural networks is processed by the following anti-normalization function to get the actual output.

$$x = x^* * (x_{ab} - x_{me}) + x_{me} \tag{61}$$

4) LOSS FUNCTION

Mean-square error function is employed to be the loss function in the learning process of BP networks. For a batch of P samples, the loss function is shown as equation 62.

$$E = \frac{1}{p} \sum_{i=1}^{p} (y_i - y_{tri})^2 \tag{62}$$

where, P is the quantity of each training batch. y_{tri} is the standard output of the training sample i, and y_i is the actual output of the neural network with the same training sample.

5) WEIGHT UPDATE STRATEGY

According to the theory of back propagation, the weight gradient of the output layer is calculated as follows.

$$\begin{aligned}\Delta w_{jk}^2 &= -\eta\frac{\partial E}{\partial w_{jk}^2} = -\eta\frac{\partial}{\partial w_{jk}^2}\left(\sum_{p=1}^{P} E_p\right)\\ &= \sum_{p=1}^{P}\left(-\eta\frac{\partial E_p}{\partial w_{jk}^2}\right) = \sum_{p=1}^{P}\left(-\eta\frac{\partial E_p}{\partial y_j} * \frac{\partial y_j}{\partial w_{jk}^2}\right)\\ &= \sum_{p=1}^{P}\sum_{j=1}^{m}\eta\left(y_i^p - y_{tri}^p\right) * z_k \end{aligned} \tag{63}$$

The weight gradient of the hidden layer is calculated as follows.

$$\begin{aligned}\Delta w_{ij}^1 &= -\eta\frac{\partial E}{\partial w_{ij}^1} = -\eta\frac{\partial}{\partial w_{ij}^1}\left(\sum_{p=1}^{P} E_p\right)\\ &= \sum_{p=1}^{P}\left(-\eta\frac{\partial E_p}{\partial w_{ij}^1}\right) = \sum_{p=1}^{P}\left(-\eta\frac{\partial E_p}{\partial S_j} * \frac{\partial S_j}{\partial w_{ij}^1}\right)\\ &= \sum_{p=1}^{P}\sum_{k=1}^{n_{ou}}\eta\left(y_k^p - y_{trk}^p\right) * w_{jk}^2 * f'(S_j) * x_i \end{aligned} \tag{64}$$

The Adam optimizer is proposed by Kingma and Lei ba. It can automatically update the learning rate and has high computational efficiency. In this study, Adam weight update strategy is employed to update. The Adam weight update formula is shown as equation 65.

$$w_t = w_{t-1} - \alpha * \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{65}$$

where, $\hat{m}_t$ is the first moment vector of the gradient.

$$\hat{m}_t = \frac{\beta_1 * m_{t-1} + (1-\beta_1) * g_t}{1-\beta_1^t} \tag{66}$$

$\hat{v}_t$ is the second moment vector of the gradient.

$$\hat{v}_t = \frac{\beta_2 * v_{t-1} + (1-\beta_2) * g_t^2}{1-\beta_2^t} \tag{67}$$

where, g_t is the Gradient vector of step t.

D. ONLINE PATH PLANNING SYSTEM AIMING AT VIBRATION SUPPRESSION

This section details how to use PSO optimization search algorithm based on vibration evaluation function to train the BPNN solver, and how to use the trained BPNN solver to build an online path generator.

1) TRAINNING FRAMEWORK

As shown in Figure 12, the samples inputs for training are inputted to PSO searcher and BPNN solver respectively. PSO searcher uses vibration evaluation function as the objective function to search out the optimal trajectory interpolation points vectors, and BPNN network generates adjustment vectors. Due to the different meanings of the output values of the two methods, the optimal trajectory interpolation points vectors should be firstly subtracted from the initial interpolation points vectors solved by quintic interpolation to obtain the optimal interpolation points vectors. The loss function is calculated by the error between the optimal interpolation point and the interpolation point generated by BPNN solver, and the BPNN solver is adjusted by weight update strategy. The training process should be finished after enough iterations until the loss function is small enough. The trained BPNN can replace PSO search algorithm to generate interpolation point vectors with small enough vibration.

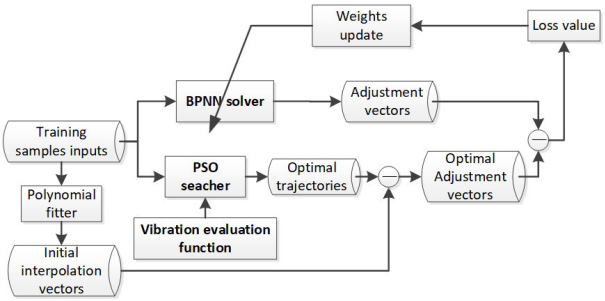


FIGURE 12. Training framework of algorithm.

The number of training samples should be sufficient and evenly distributed in the range of motion of the manipulator. Increasing the number of samples can improve the learning effect of BPNN solver, but obviously it will also increase the learning time.

2) ONLINE TRAJECTORY PLANNING SYSTEM

The online path generator composed of trained BPNN network is shown in Figure 13. The initial interpolation point vector is generated by quintic polynomial interpolation, and the adjustment vector is generated by neural network solver. The final interpolation point data is obtained by the superposition of initial interpolation point vector and adjustment vector, and the final trajectory equation is generated by quintic polynomial fitting.

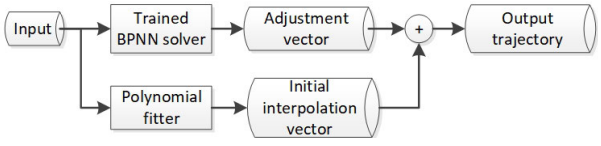


FIGURE 13. Algorithm framework.

IV. SIMULATION

We consider a 3-DOF robotic manipulator as shown in Figure 14. The robot has three rotational degrees of freedom in a plane. The rotation angel θ of all connecting links is limited to $[-3/4pi, 3/4pi]$. The parameters of the manipulator

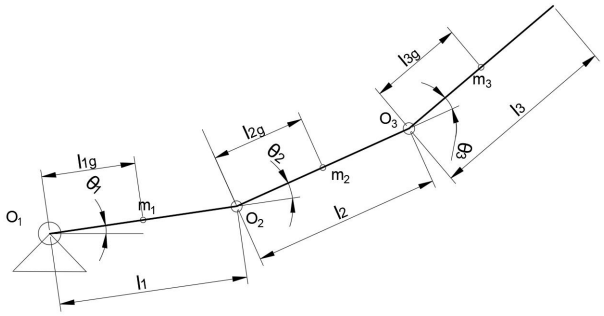


FIGURE 14. Model of 3-DOF robotic manipulator.

are shown as equation 68.

$$\begin{cases} l_1 = 2, l_2 = 1, l_3 = 1 \\ m_1 = 100, m_2 = 50, m_3 = 250 \\ l_{1g} = 0.5 * l_1, l_{2g} = 0.5 * l_2, l_{3g} = L_3 \\ I(1) = 300e{-}8, I(2) = 200e{-}8, I(3) = 100e{-}8 \\ E = 0.7 * 10^{11} \end{cases} \tag{68}$$

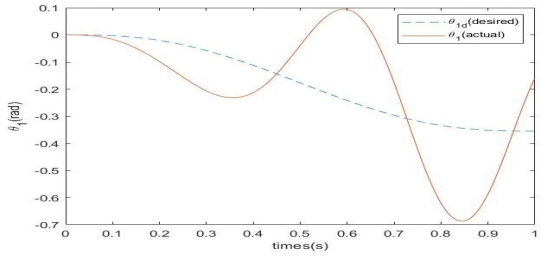
In order to reduce the calculation amount of this simulation, the motion of each joint is limited to the motion from the static state of 0 degree angle to the static state of any angle within 1 second, that is, the angle, angular velocity, angular acceleration and time of the initial state of each revolute joint are set to 0, and the angle of the end point is randomly selected between [−3 / 4pi, 3 / 4pi]. The angular velocity and angular acceleration of the end state of each revolute joint are also set to 0, and the end time is set to 1. This kind of motion is a special state of the manipulator movement, but it can reflect the vibration suppression effect of the manipulator in the general movement process, so it is representative in the simulation verification.

A. QUINTIC POLYNOMIAL FUNCTION TRAJECTORY PLANNING

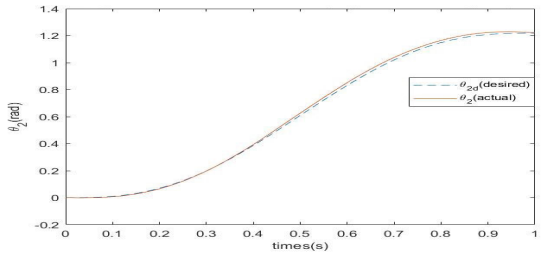
The smooth motion trajectory can effectively reduce the structural vibration caused by the sudden change of speed and acceleration, therefore, high-order polynomial smooth function path planning is a common method of vibration suppression.

In this method, the start and end points are connected with the curve of quintic polynomial function, so as to ensure the continuous derivative of velocity and acceleration in the motion trajectory. The specific process of this method has been given in Section III.B.4 (Initialization of search variables). Figure 15–18 illustrate the simulation results.

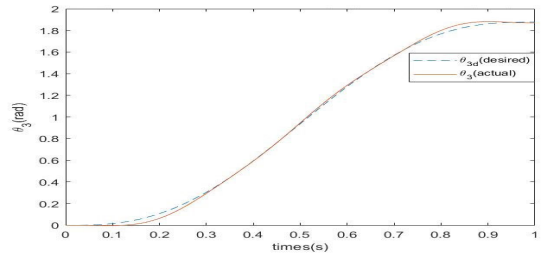
From Figure 15, we can see that the angle output curve of each joint can track the expected curve but there are some errors. Figure 16 shows that the tracking error is within 0.35 degree. Figure 17 shows that the trajectory of manipulator end can track the expected curve but there are some errors. Figure 18 shows that the tracking error of the end point position of the manipulator is within 0.8m.



(a) Joint 1



(b) Joint 2



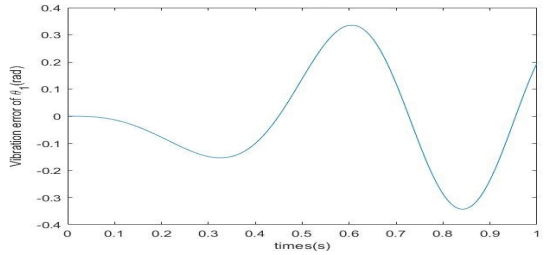
(c) Joint 3

FIGURE 15. Vibration performance of joint under QP method.

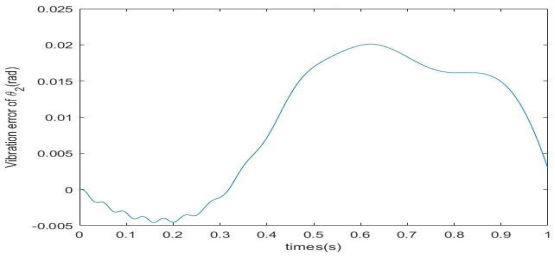
B. TRAJECTORY PLANNING BASED ON A PSO METHOD

For the simulation of a PSO method, we set the number of interpolation points per degree of freedom $n = 3$, so the number of optimized variables $D = 3{*}n{*}m = 27$. Set the size of the particle swarm $N = 99$; set the number of iterations $MaxDT = 80$; set the inertia weight $w = 0.7298$; set the learning factor $c1 = 1.49618, c2 = 1.49618$. Figure 19–22 illustrate the simulation results.

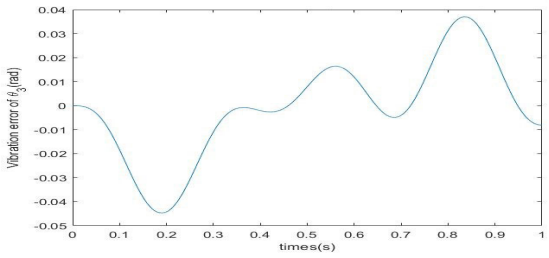
From Figure 19, it can be seen that the tracking accuracy of joint 1 curve is significantly higher than that of the QC method; but the tracking accuracy of joint 2 and 3 curve is slightly lower than that of the QC method, that is because the residual energy generated by vibration of joint 1 is much larger than that of joint 2 and joint 3, and the optimization algorithm will preferentially suppress the vibration of joint 1. It can be seen from Figure 20 that the tracking error is below 0.14 rad. Figure 21 shows that the end tracking effect of the manipulator is significantly improved compared with the QC method. Figure 22 shows that the horizontal vibration at the end of the manipulator is kept within 0.25, which is significantly lower than the 0.5 vibration error of the QC method; the vertical vibration at the end of the manipulator is kept within 0.55, which is lower than the 0.65 vibration error of the QC method.



(a) Joint 1

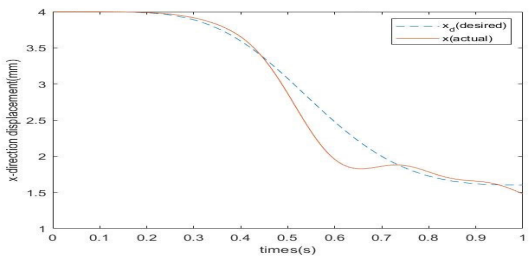


(b) Joint 2

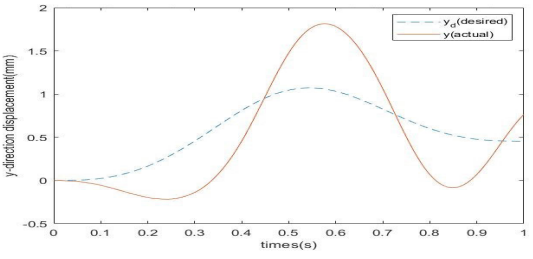


(c) Joint 3

FIGURE 16. Vibration error of joint under QP method.

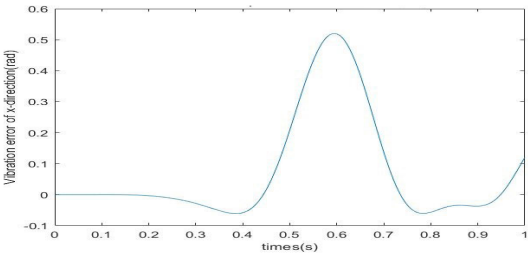


(a) X-coordinate

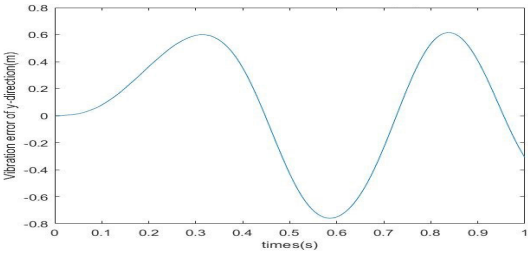


(b) Y-coordinate

FIGURE 17. Vibration trajectory of manipulator end under QC method.

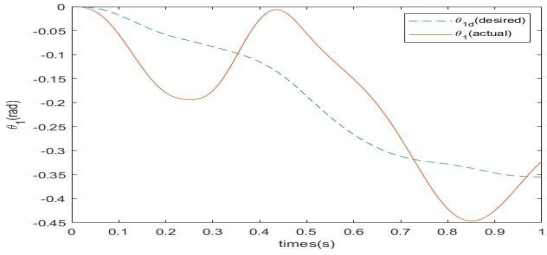


(a) X-coordinate

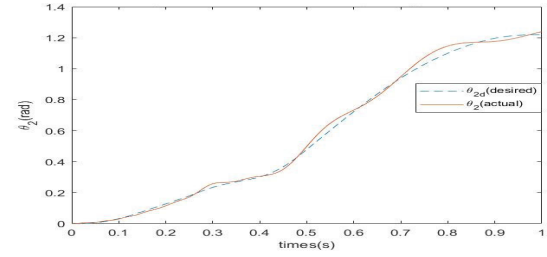


(b) Y-coordinate

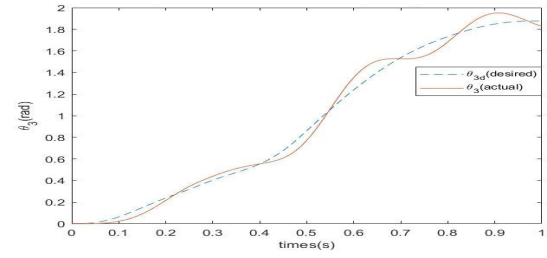
FIGURE 18. Vibration error of manipulator end under QC method.



(a) Joint 1



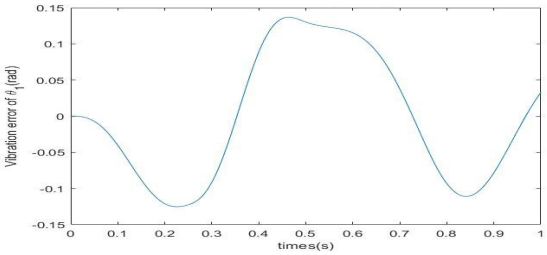
(b) Joint 2



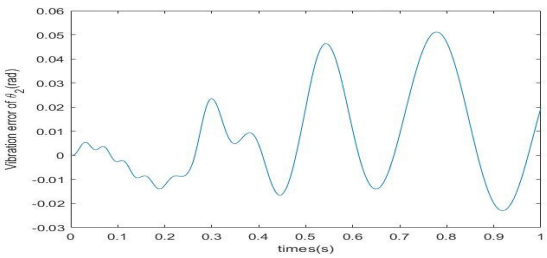
(c) Joint 3

FIGURE 19. Vibration performance of joint under PSO method.

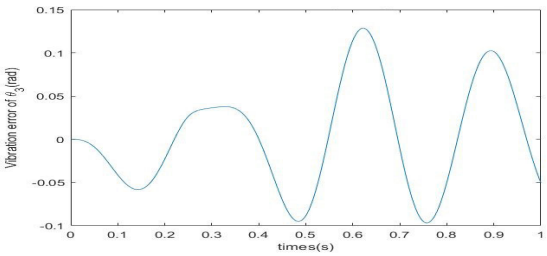
However, the trajectory planning method based on particle swarm optimization has a huge amount of computation, and it takes 6-8 hours to complete a trajectory planning process, so it is impossible to carry out on-line trajectory planning in the process of motion.



(a) Joint 1

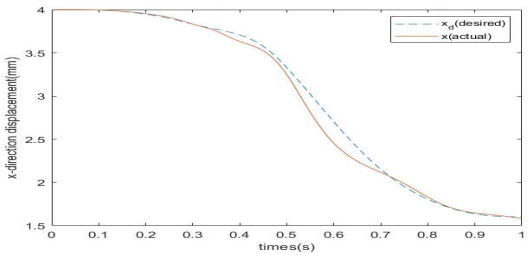


(b) Joint 2

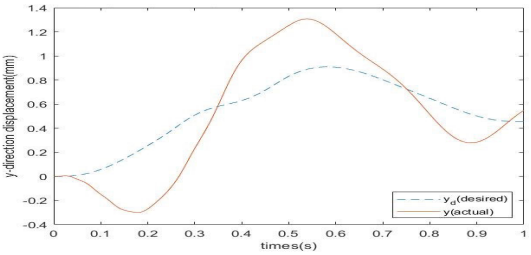


(c) Joint 3

FIGURE 20. Vibration error of joint under PSO method.



(a) X-coordinate

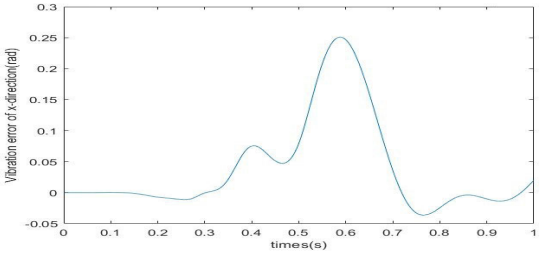


(b) Y-coordinate

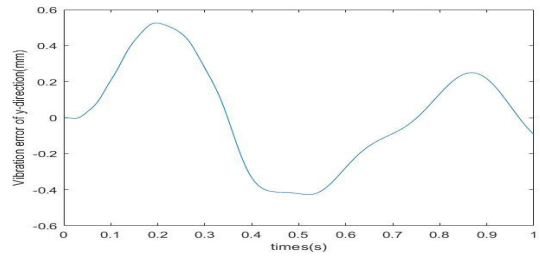
FIGURE 21. Vibration trajectory of manipulator end under PSO method.

C. TRAJECTORY PLANNING BASED ON A BPNN

For the BPNN, 256 nodes are used for the NN. The training samples of PSO are used as the training set



(a) X-coordinate



(b) Y-coordinate

FIGURE 22. Vibration error of manipulator end under PSO method.

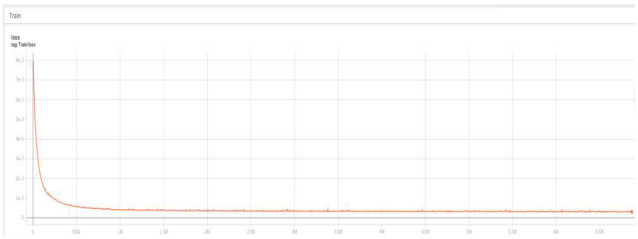
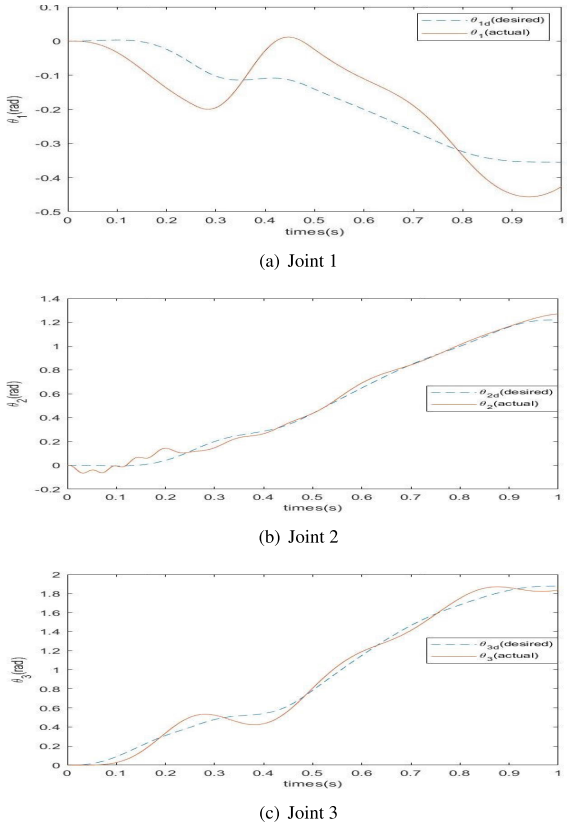


FIGURE 23. Loss-t curve of BPNN training.

of BPNN. In order to make the sample data evenly distributed, we take 343 points evenly distributed in domain of $[-3/4 * pi, 3/4 * pi] \times [-3/4 * pi, 3/4 * pi] \times [-3/4 * pi, 3/4 * pi]$. In addition, another 300 randomly distributed points are selected also in above domain. Figure 23 show the loss value of neural networks training process. Pick a random sample not included in the training samples, Figure 24–27 illustrate the simulation results.

From Figure 23, it can be seen that the loss value of neural networks training process is effectively decreases and stabilizes around $5e-4$, which means that neural networks can effectively learn from training samples.

From Figure 24, it can be seen that the tracking accuracy of joint 1 curve is significantly higher than that of the QC method; but the tracking accuracy of joint 2 and 3 curve is slightly lower than that of the QC method, that is because the residual energy generated by vibration of joint 1 is much larger than that of joint 2 and joint 3, and the optimization algorithm will preferentially suppress the vibration of joint 1. It can be seen from Figure 25 that the tracking error is below 0.14 rad. Figure 26 shows that the end tracking effect of the manipulator is significantly improved compared with the the QC method. Figure 27 shows that the horizontal vibration

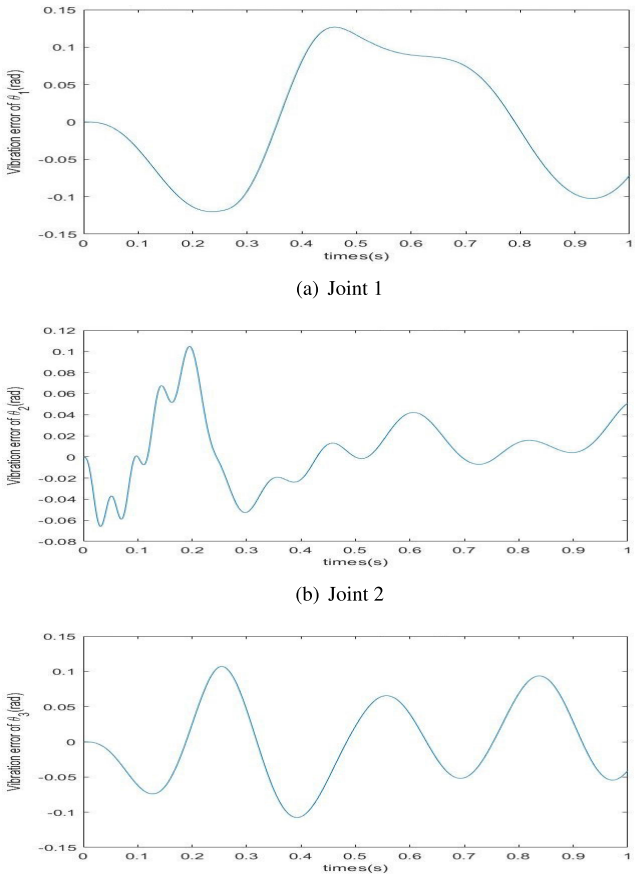


(a) Joint 1

(b) Joint 2

(c) Joint 3

FIGURE 24. Vibration performance of joint under BPNN method.



(a) Joint 1

(b) Joint 2

(c) Joint 3

FIGURE 25. Vibration error of joint under BPNN method.

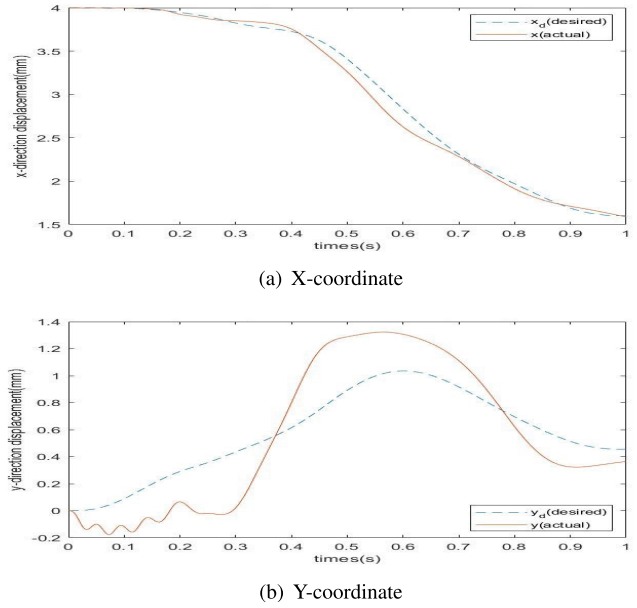
at the end of the manipulator is kept within 0.25, which is significantly lower than the 0.5 vibration error of the QC method; the vertical vibration at the end of the manipulator is kept within 0.5, which is lower than the 0.65 vibration error of the QC method.

D. RESULT COMPARISON

In order to further elaborate the advantages of the BPNN based path planning method, the tracking errors of the above three methods are compared in Figure 28,29. In order to ensure the universality of simulation, 20 random samples are further simulated. Based on the vibration residual energy of the QC method, the vibration suppression rates of these 20 samples under the PSO and the BPNN trajectory planning method is shown in Figure 30. The average program running time and vibration suppression ratios (based on the QC method) of 20 samples under three methods are shown in Figure 31.

From Figure 28,29, we can see that the tracking accuracy of BPNN method is similar to that of the PSO method, which is significantly smaller than that of the QC method. The average running time of the BPNN method is 7.4 seconds, which is similar to the QC method(4.6s), but significantly shorter than that of the PSO method(24480s).

It can be seen from Figure 30 that the BPNN trajectory planning method has obvious vibration suppression effect



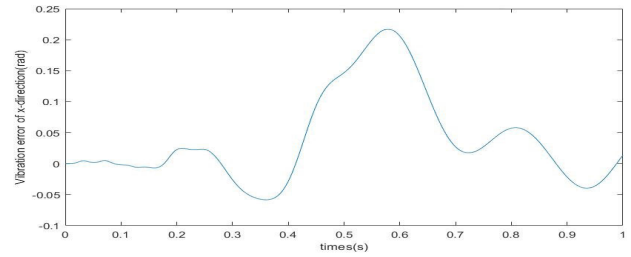
(a) X-coordinate

(b) Y-coordinate

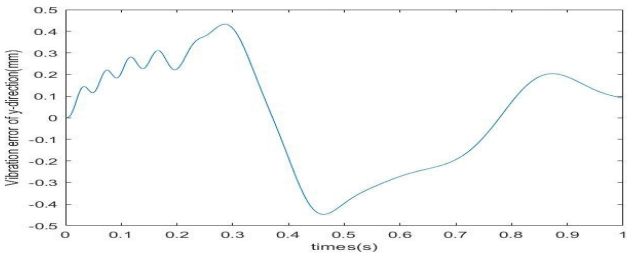
FIGURE 26. Vibration trajectory of manipulator end under BPNN method.

compared with the QC method, but slightly lower than the PSO method.

It can be seen from Figure 31 that the average running time of the BPNN method is 7.4 seconds, which is similar to

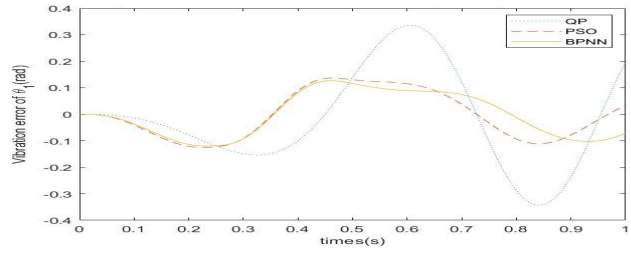


(a) X-coordinate

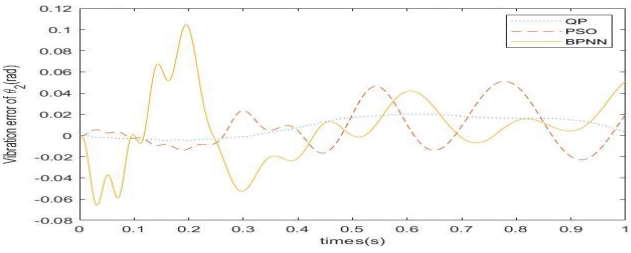


(b) Y-coordinate

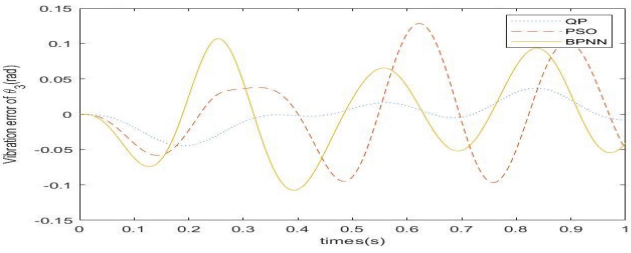
FIGURE 27. Vibration error of manipulator end under BPNN method.



(a) Joint 1

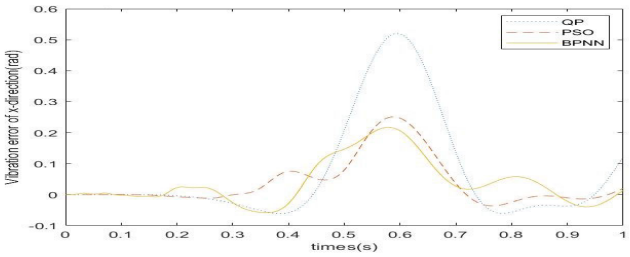


(b) Joint 2

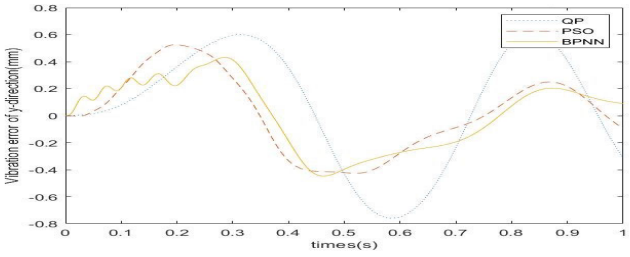


(c) Joint 3

FIGURE 28. Comparison of joint vibration errors under three method.



(a) X-coordinate



(b) Y-coordinate

FIGURE 29. Comparison of vibration errors at manipulator end under three method.

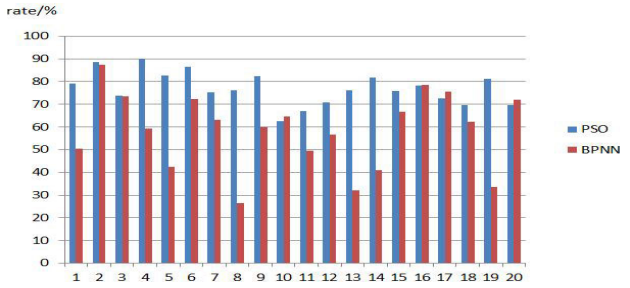


FIGURE 30. Vibration suppression rates of 20 samples based on QC method.

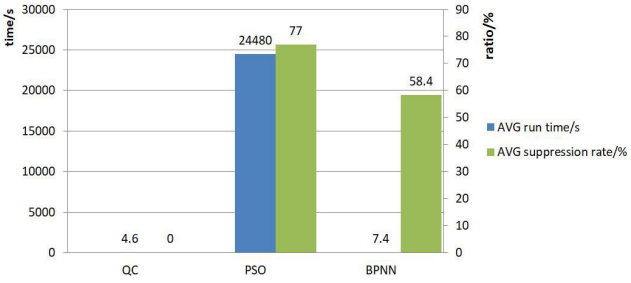


FIGURE 31. Average performance comparison of three methods.

the QC method(4.6s) and significantly shorter than the PSO method(24480s), but the vibration suppression rate of BPNN method based on traditional QC method is 58.4%, which is similar with PSO method(77%). It should be noted that the running time is the performance on personal computer, which can be further reduced on high performance industrial computer. To sum up, the BPNN trajectory planning method has good vibration suppression effect and short program running time, and can be used for on-line operation.

V. CONCLUSION

The vibration dynamic equation and the simplified analytical solution method are established for a planar felxible-link manipulator, and the vibration evaluation function based

on the residual energy is established. Based on a BPNN, the vibration suppression trajectory planning method is established, and an off-line training system for BPNN solver is established by PSO algorithm based on vibration evaluation function. The simulation results show that this method has excellent vibration suppression effect and rapid running speed, and can be used for on-line planning of vibration suppression trajectory in the process of manipulator motion.

This method is only applied to planar flexible-link robots, and the vibration dynamic equation of three-dimensional space robot is more complex, so the vibration evaluation method needs to be further studied. In addition, it is planned to build a physical prototype of the flexible-link robot system to further verify the effectiveness of the vibration suppression method. Futher more, the adaptive control method of the manipulator will be studied to compensate the flexible manipulator system with unknown dynamic parameters.

REFERENCES

[1] M. Yamano, J.-S. Kim, and M. Uchiyama, "Hybrid position/force control of two cooperative flexible manipulators working in 3D space," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 1998, pp. 1110–1115.

[2] F. Matsuno and M. Hatayama, "Quasi-static cooperative control of two two-link flexible manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 1995, pp. 615–620.

[3] C. F. Zhang and L. L. Han, "Preliminary study on lunar robot for manned lunar exploration," *Manned Spaceflight*, vol. 25, no. 5, pp. 561–571, 2019.

[4] A. G. Kelkar and S. M. Joshi, "Capture and control of unknown space objects with flexible multi-link manipulators," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 1605–1606.

[5] S. Ananthakrishnan and N. M. Wahbah, "Control of flexible space station remote manipulator training system," in *Proc. IEEE Int. Conf. Control Appl.*, Sep. 1993, pp. 15–20.

[6] Z. Xiaojin, L. Meiyu, C. Hao, and Z. Xiaoyu, "3D reconstruction method of space manipulator vibration shape based on structural curvatures," in *Proc. 8th Int. Conf. Electron. Meas. Instrum.*, 2007, pp. 772–775.

[7] P. T. L. M. van Woerkom and A. K. Misra, "Robotic manipulators in space: A dynamics and control perspective," *Acta Astronautica*, vol. 38, nos. 4–8, pp. 411–421, 1996.

[8] G. Jorgensen and E. Bains, "SRMS history, evolution and lessons learned," in *Proc. AIAA SPACE Conf. Expo.*, Sep. 2011, p. 7277.

[9] Y. Li, S. S. Ge, Q. Wei, D. Zhou, and Y. Chen, "Modular robotic system for nuclear decommissioning," in *Proc. Int. Conf. Social Robot.*, 2018, pp. 297–307.

[10] S. Shi, Y. Cheng, H. Pan, W. Zhao, and H. Wu, "Development and error compensation of a flexible multi-joint manipulator applied in nuclear fusion environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3587–3592.

[11] H. Su, J. Sandoval, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, "Safety-enhanced collaborative framework for tele-operated minimally invasive surgery using a 7-DoF torque-controlled robot," *Int. J. Control, Autom. Syst.*, vol. 16, no. 6, pp. 2915–2923, Dec. 2018.

[12] 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.

[13] W. He and S. S. Ge, "Vibration control of a flexible beam with output constraint," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5023–5030, Aug. 2015.

[14] W. He and S. S. Ge, "Vibration control of a flexible string with both boundary input and output constraints," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1245–1254, Jul. 2015.

[15] S. S. Ge, T. H. Lee, and G. Zhu, "Energy-based robust controller design for multi-link flexible robots," *Mechatronics*, vol. 6, no. 7, pp. 779–798, Oct. 1996.

[16] T. Li, "Research on flexible joint robot and its kinematic calibration and viration suppression research," (in Chinese), Ph.D. dissertation, Harbin Inst. Technol., Harbin, China, 2012.

[17] Y. T. Yang, "Dynamic modeling trajectory planning and vibration suppressing of the spatial flexible manipulator," (in Chinese), Ph.D. dissertation, Beijing Inst. Technol., Beijing, China, 2014.

[18] K. F. Guo, "Study on dynamic modeling and elastic vibration control of flexible manipulator," (in Chinese), M.S. thesis, Zhengzhou Univ., Zhengzhou, China, 2017.

[19] W. He, S. S. Ge, and J. Zhang, "Dynamic modeling and system identification for the lower body of a social robot with flexible joints," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2011, pp. 342–347.

[20] X. Zhang, X. Wang, J. K. Mills, and W. L. Cleghorn, "Dynamic modeling and active vibration control of a 3-PRR flexible parallel manipulator with PZT transducers," in *Proc. 7th World Congr. Intell. Control Automat.*, 2008, pp. 461–466.

[21] G. Chen, D. Liu, and L. Zhang, "Impact dynamic modeling of space flexible manipulators based on continuous approach," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 1953–1957.

[22] H. A. Talebi, R. V. Patel, and H. Asmer, "Dynamic modeling of flexible-link manipulators using neural networks with application to the SSRMS," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, Oct. 1998, pp. 673–678.

[23] H. S. Yue, D. Henrich, and W. L. Xu, "Trajectory planning in joint space for flexible robots with kinematics redundancy," in *Proc. IASTED Int. Conf.*, 2001, pp. 1–6.

[24] B. Cao, K. Sun, T. Li, Y. Gu, M. Jin, and H. Liu, "Trajectory modified in joint space for vibration suppression of manipulator," *IEEE Access*, vol. 6, pp. 57969–57980, 2018.

[25] V. Raju, D. Maheswari, and S. K. Patnaik, "Active vibration control of piezo actuated cantilever beam using PSO," in *Proc. IEEE Students' Conf. Electr., Electron. Comput. Sci.*, Mar. 2012, pp. 1–5.

[26] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2943–2949, Apr. 2020.

[27] W. Qi, H. Su, and A. Aliverti, "A smartphone-based adaptive recognition and real-time monitoring system for human activities," *IEEE Trans. Human-Machine Syst.*, early access, Apr. 24, 2020, doi: 10.1109/THMS.2020.2984181.

[28] Q. Zhu and S. Mao, "Inertia parameter identification of robot arm based on BP neural network," in *Proc. 33rd Chin. Control Conf.*, Jul. 2014, pp. 6605–6609.

[29] N. Zhang, Y. Zhang, J. Cheng, and C. Ma, "Inverse kinematics solution for six-DOF serial robots based on BP neural network," in *Proc. Chin. Automat. Congr. (CAC)*, Oct. 2017, pp. 1154–1157.

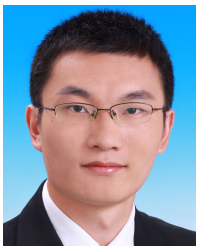
[30] Y. Zhou, W. Tang, and J. Zhang, "Algorithm for multi-joint redundant robot inverse kinematics based on the Bayesian–BP neural network," in *Proc. Int. Conf. Intell. Comput. Technol. Automat. (ICICTA)*, Oct. 2008, pp. 173–178.

[31] X. Dai, J. Wang, and J. Zhao, "Research on multi-robot task allocation based on BP neural network optimized by genetic algorithm," in *Proc. 5th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2018, pp. 478–481.

[32] N. Yi, W. Xu-Hao, and H. Lili, "Force feedback predictive control based on BP neural network of MIS robot," in *Proc. Int. Conf. Electr. Inf. Control Eng.*, Apr. 2011, pp. 419–422.

YUANYUAN LI received the B.S. and M.S. degrees from Southwest Jiaotong University, Chengdu, China, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu. His current research interest includes robot systems and its control theory.

SHUZHI SAM GE received the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986, and the Ph.D. degree from the Imperial College London, London, U.K., in 1993. He is currently the Director of the Social Robotics Laboratory of Interactive Digital Media Institute, Singapore, and the Centre for Robotics, Chengdu, China, and a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, on leave from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu. He has coauthored 4 books and over 300 international journal articles and conference papers. His current research interests include social robotics, adaptive control, intelligent systems, and artificial intelligence. He has served as a member of the Board of Governors at the IEEE Control Systems Society, from 2007 to 2009. He is a Fellow of the International Federation of Automatic Control, the Institution of Engineering and Technology, and the Society of Automotive Engineering. He is the Editor-in-Chief of the *International Journal of Social Robotics* (Springer). He has served/been serving as an Associate Editor for a number of flagship journals, including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Automatica*. He serves as a Book Editor for the Taylor and Francis Automation and Control Engineering Series. He has served as the Vice President for Technical Activities, from 2009 to 2010, and the Membership Activities, from 2011 to 2012.



QINGPING WEI received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2010, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, Beijing, China, in 2015. His current research interests include nuclear robot and industrial automation.

TAO GAN received the B.S. degree from the Chengdu University of Technology, Sichuan, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include reinforcement learning and computer vision.

XIAOLIN TAO received the B.S. degree from Southwest Minzu University, Chengdu, China, in 2018. She is currently pursuing the master's degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu. She is also researching on reinforcement learning and transfer learning.