中國機械工程學刊第四十五卷第六期第 605~618 頁(民國一百一十三年)
Journal of the Chinese Society of Mechanical Engineers, Vol.45, No.6, pp605~618 (2024)

Optimization of Control Parameters for Robot Integrated Joint Modules Based on an MA-BPNN Model

Qiu-Shi Hu*, Heng Li*, Hao-Qiang Li*, Guo-Wei Wang* and Lei Li*

Keywords : joint modules, responsiveness performance, control parameters, back-propagation neural network.

ABSTRACT

This study introduces a method to predict the responsiveness performance and optimize the control parameters of joint modules. An integrated joint module was created to assess overshoot and response time under various operating conditions and control parameter values. The results reveal nonlinear impact of conditions and parameters on the responsiveness performance, indicating a complex multifactorial issue. An MA-BPNN model was then established by merging the BPNN model with the MA algorithm, offering improved prediction accuracy and computational efficiency compared to traditional models. The application of Latin hypercube sampling and the MA-BPNN model yielded the Pareto optimal solutions for the control parameters of the joint module under commonly known operating conditions. Notably, higher load and joint speed values require a suitably higher proportional gain coefficient in order to meet the system demands. These findings hold significant value for the design and optimization of actual joint modules.

INTRODUCTION

Robots play a crucial role in various fields, such as intelligent manufacturing, medical devices and aerospace (Zhao et al., 2023; Zhen et al., 2022; Cai et al., 2019). The joint modules, as the core components of robots, exhibit compact structure, modular design, lightweight, high precision and safety reliability (Hu et al., 2021).

These joint modules drive the robot to achieve rotational motion, while their design and performance directly affect the motion control quality and accuracy of the robot system. Therefore, optimizing the design and performance of the joint modules can further improve the motion control accuracy, stability and responsiveness of the robot, enabling it to meet the demanding requirements of different fields (Huang et al., 2021).

Various methods, such as analytical modeling, experimental testing and machine learning, can be used to predict the performance of robot joint modules (Tian et al., 2023). Analytical modeling involves comprehensive analysis of the joint module structure, its dynamic characteristics and its working environment, in order to establish corresponding mathematical models, used for performance prediction (Cui et al., 2022). However, analytical modeling faces challenges in the case of complex nonlinear systems. Experimental testing involves testing and measurement in real life environment, in order to collect performance data of joint modules, under different working conditions, followed by analysis and processing (Bittencourt et al., 2014). This approach can obtain reliable performance data under real working conditions, but testing costs and time investment need to be considered. Machine learning methods utilize a large amount of known data to train models, recognize patterns and make predictions. They are suitable for handling complex nonlinear systems and show strong flexibility and adaptability, allowing them to learn features and patterns from data (Jiang et al., 2023).

In terms of precise motion control of robotic joint modules, control algorithms and parameter tuning play a critical role (Nohooji et al., 2024; Zhang et al., 2023; Ang et al., 2005). In the past, Proportional-Integral-Derivative (PID) control was most commonly used, due to its superior performance and ease of implementation. However, traditional PID control methods face difficulties in parameter tuning, lack of adaptability and insufficient robustness, when dealing with complex nonlinear control problems in robotics, often failing to meet real control requirements (Angel et al., 2018). In order to address these issues, researchers have proposed more advanced control algorithms, such as

Paper Received March, 2024. Revised May, 2024. Accepted May, 2024. Author for Correspondence: Qiu-Shi Hu.

* *School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China.*

fuzzy control, sliding mode control, adaptive control and neural network control algorithms (Zhong et al., 2023; Nikdel et al., 2017; Baek et al., 2016). Méndez et al. (2020) proposed a dynamic adjustment of PID gains, using three interval type-1 non-singleton type-2 fuzzy logic systems, while parameter tuning is done based on a non-singleton backpropagation algorithm, to achieve improved performance and stable behavior in the control system. Han et al. (2023) presented a fuzzy gain scheduling PID controller, based on the dynamic characteristics of a hybrid robot, named TriMule. By deducing control parameters online, the impact of dynamic characteristics on the control system can be significantly reduced. Van et al. (2020) developed a self-tuning fuzzy mechanism and an approximate technique, based on time delay estimation, resulting in a novel control method, called self-tuning fuzzy PID-nonsingular fast terminal sliding mode control. This method enhances transient response and provides reduced overshoot and steady-state error. Liu et al. (2020) proposed a robust adaptive control strategy, combining fuzzy logic system and sliding mode control, whereas numerical simulations demonstrated system stability and validated the effectiveness and robustness of the approach. Carlucho et al. (2020) presented an intelligent control system for mobile robots that leverages deep reinforcement learning techniques, to facilitate self-adaptation of multiple PID controllers. The approach permits adjustment of PID controller parameters or gains, enhancing the robustness and control efficiency of the robotic control system. Son et al. (2017) established a control system that integrates an adaptive feed-forward neural controller and a PID controller for controlling the joint-angle position of a SCARA parallel robot. The parameters of the controller are automatically tuned online, during the control process, using an adaptive back-propagation algorithm. Zheng et al. (2023) proposed an adaptive memetic differential evolution-backpropagation-fuzzy neural network control method, using adaptive differential evolution for global parameter optimization and implementing online optimization, using backpropagation algorithm.

Among these control approaches, neural network control algorithms stand out with some advantageous characteristics (Liang et al., 2021; Jian et al., 2019). They possess strong nonlinear approximation capabilities, which allow for accurate modeling of the nonlinear dynamic characteristics of complex systems. Additionally, neural networks have adaptive learning capabilities, parallel processing abilities and can serve well generalization scenarios. Thereby, they can improve control performance, efficiency, response speed, as well as adapt better to unknown operating or environmental conditions. The selection of suitable training algorithms is crucial for the training process of neural networks (Wang et al., 2022). Rahideh et al. (2008) proposed a neural network-based method for nonlinear dynamics modeling of a dual-rotor multi-input multi-output system, utilizing the Powell-Beale version of conjugate gradient and scaled conjugate gradient learning algorithms to train the feedforward neural network. The trained neural network model can effectively capture the highly nonlinear characteristics of the system, making it suitable for the development of complex controllers. Wang et al. (2018) used an improved parallel Levenberg-Marquardt (LM) algorithm to train recursive neural networks, in order to improve the learning convergence speed of the recursive neural network in robot control systems and reduce control latency. Zuo et al. (2019) used a neural network trained with back-propagation (BP) to compensate for errors in the external force detection model of a humanoid robotic arm, in order to obtain more precise external force values for the robotic arm.

Back-propagation neural networks (BPNNs) demonstrate outstanding capabilities in nonlinear analysis, generalization, and fault tolerance, especially in situations requiring flexibility and adaptation to large-scale data. However, BPNNs may have slower convergence speeds and carry the risk of falling into local optimal solutions. Regarding the search for global optimal solutions, metaheuristic algorithms demonstrate powerful capabilities and excel in various complex problems (Lin et al., 2020). Ning et al. (2022) proposed a small obstacle size prediction method based on a genetic algorithm-back propagation (GA-BP) neural network, used for quantitative perception of the environment by mobile robots. Bai et al. (2021) proposed a dynamic weight particle swarm optimization-based sine map (SDWPSO) method for optimizing the BPNN algorithm. The proposed SDWPSO-BPNN model has been applied to reliability prediction in turbocharger and industrial robot systems, leading to significantly improved prediction accuracy compared to commonly used reliability prediction methods. Shirzadeh et al. (2021) proposed an adaptive robust controller based on type-2 fuzzy neural network (T2FNN), which utilized the cuckoo optimization algorithm (COA) to optimize the relevant parameters of the controller, achieving the tracking of desired trajectories for a quadrotor. Yan et al. (2024) used a BP neural network model optimized by the Sparrow Search Algorithm (SSA) to predict the clamping force of a minimally invasive surgical robot system.

These metaheuristic algorithms optimization algorithms have demonstrated good performance in enhancing the performance of neural networks and avoiding falling into local optima. In order to effectively avoid local optima, simplify the parameter selection process, and enhance the ability to handle outliers, ultimately improving the predictive accuracy

and robustness of neural networks. This study proposes a method based on the mayfly algorithm (MA) and the BPNN, for responsiveness performance prediction and control parameters optimization in the joint module. The remaining content of the article is organized as follows. In Section 2, an integrated joint module is developed and tested, while the influence of operating conditions and control parameters, on its responsiveness performance, is discussed. In Section 3, an MA-BPNN model is established for predicting the responsiveness performance of the joint module, whereas its superiority is validated based on comparisons to the BPNN and PSO-BPNN models. In Section 4, the optimization of control parameters of the joint module, under common operating conditions, is carried out. Finally, the main conclusions are summarized in Section 5. The proposed MA-BPNN model and the optimization method for control parameters of the joint module provide important guidance and reference for the design of control systems, the improvement of control performance and the optimization of system stability.

RESPONSIVENESS PERFORMANCE TESTS OF THE JOINT MODULE

This study is about the development of an integrated joint module, where the goal is to simplify system integration, enhance system performance, provide convenient control interfaces and improve intelligent capabilities. The responsiveness performance plays a critical role in the efficiency, accuracy and stability of the joint module. Responsiveness performance tests, conducted under various operating conditions and control parameter settings, provide useful data sets that serve as foundation for control parameters optimization methods.

Development of the Integrated Joint Module

The main hardware components, including a harmonic drive, a brushless torque motor, a drive controller, an electromagnetic brake and an angle encoder, were designed and assembled to form a joint module with driving and control functions provided through real-time communication with a main computer unit.

The development process for the joint module is illustrated in Fig. 1 and can be described in detail as follows:

(1) Hardware structural design. A model SHG-14-100 harmonic drive was selected to improve the transmission accuracy of the joint module. Additionally, improvements were made to the wave generator structure, by increasing its axial dimensions, allowing direct coupling with the electromagnetic brake and reducing assembly errors.

(2) Hardware selection. The technical features of the designed joint module are an outer diameter of 70 mm, output torque of 50 Nm and transmission accuracy of 10". The main hardware components were chosen as follows: a brushless torque motor (TBSM60), a drive controller (STM32), an electromagnetic brake (S060GL) and an angle encoder (ECODER35).

(3) Drive control system design. The drive control system, with the STM32F103C8T6 chip at its core, operates the brushless torque motor and controls the electromagnetic brake by forwarding commands generated by the software running on the main computer. Additionally, it periodically collects data from the angle encoder, the zero-position sensor, the gyro accelerometer and the temperature sensor, transmitting these and the driver feedback data, via the controller area network-bus (CAN-BUS), to the main computer unit.

(4) Implementation and debugging. The motion control algorithm debugging was done on the main computer, in order to validate the motion capabilities of the joint module.

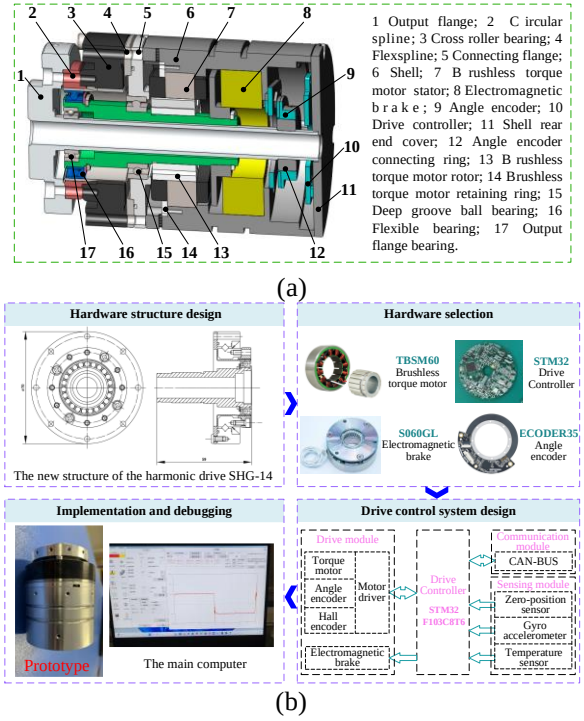


Fig. 1. The development process of the joint module: (a) the overall structure; (b) the development process.

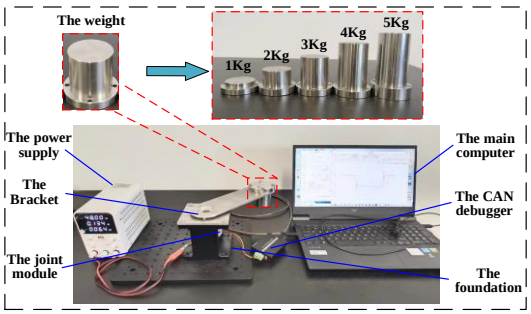
Responsiveness Performance Tests

The study of the responsiveness performance of joint modules contributes to optimizing control strategies and improving system stability, motion efficiency, and control accuracy. The overshoot (O_{shoot}) and response time (R_{time}) factors serve as critical performance indicators for joint modules and play a significant role in evaluating the dynamic

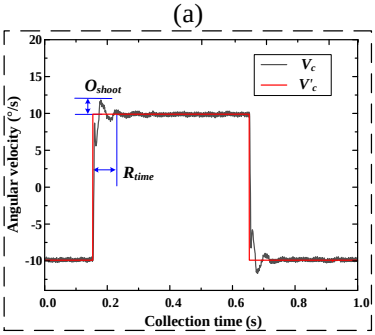
behavior of the system and in the control parameters calibration.

A testing platform was constructed, to explore the responsiveness performance (O_{shoot} and R_{time}) of the joint module, under different operating conditions, as shown in Fig. 2a. The testing platform consists of the joint module, a foundation, an adjustable regulated power supply, a main computer unit, a CAN debugger and a load. The load size can be adjusted by changing the weights. The joint module can achieve precise speed loop control, by receiving suitable commands from the software running on the main computer unit.

The time-domain curve of the output angular velocity, as derived by the testing platform, is depicted in Fig. 2b. O_{shoot} is defined as the maximum deviation of the actual output angular velocity V_c of the joint module from the ideal output angular velocity V'_c. R_{time} represents the time required for the joint module to reach a steady-state operating speed, from the start of operation.



(a)



(b)

Fig. 2. The process of the evaluation of the joint module responsiveness performance: (a) the responsiveness performance testing platform of the joint module; (b) the time-domain curve of the V_c and V'_c.

The responsiveness performance of the joint module is influenced by operating conditions and control parameters. The operating conditions include the load (W_l) and the angular velocity (V_c), while the control parameters consist of the proportional gain coefficient (k_p) and the integral gain coefficient (k_i). As the presented joint module is primarily intended for a 5 kg robot, the W_l range is set from 1 kg to 5 kg, while the V_c range is set at [3.6 °/s, 10.8 °/s]. The k_p and k_i parameters are determined based on empirical values and are selected within the ranges of [800, 1290] and [75, 131], respectively. The tests consist a total of 9 sets of V_c, k_p, and k_i values, evenly spaced within the specified ranges, as listed in Table 1.

Table 1. The responsiveness performance test parameters of the joint module.

Parameter	Numerical value
W_l (Kg)	1, 2, 3, 4, 5
V_c (°/s)	3.6, 4.5, 5.4, 6.3, 7.2, 8.1, 9.0, 9.9, 10.8
k_p	800, 860, 920, 980, 1040, 1100, 1160, 1220, 1280
k_i	75, 82, 89, 96, 103, 110, 117, 124, 131

The steps for testing the responsiveness performance of the joint module are as follows:

Step 1: Establishing connection and operation.

1) Establishing the connection between the main computer unit and the joint module, using a CAN debugger.

2) Allowing uninterrupted operation of the joint module load-free, for 5 minutes.

Step 2: Installation, configuration and data collection.

1) Installing weights.

2) Setting values for V_c, k_p and k_i.

3) Driving the motion of the joint module using the main computer.

4) Registering the values of the time-domain V_c curve on the main computer through an angle encoder.

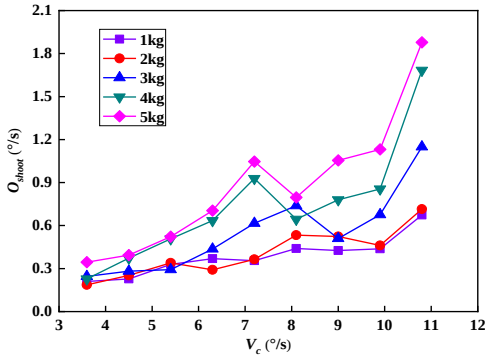
5) Recording O_{shoot} and R_{time} values for the joint module under the specific operating condition and control parameters.

Step 3: Variation of test conditions and iteration.

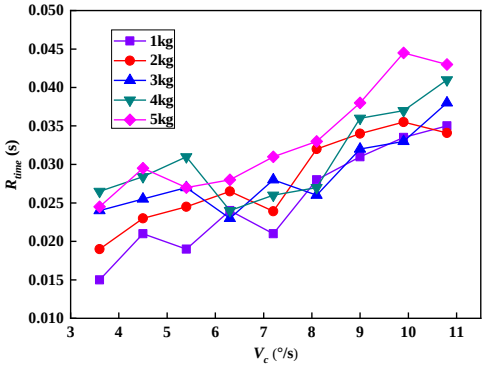
1) Change the test conditions.

2) Repeat Step 2 to obtain O_{shoot} and R_{time} values.

3) Turn off the main computer and power supply after completing all parameter values sets.



(a)



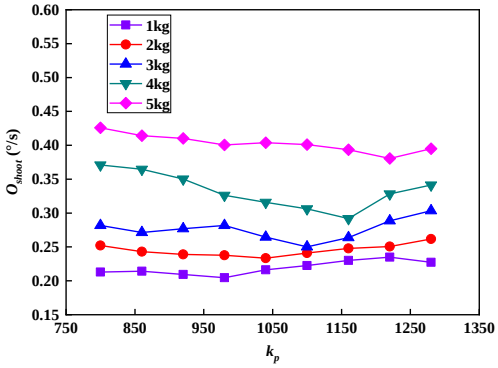
(b)

Fig. 3. The influence of varying operating conditions on the responsiveness performance of the joint module: (a) the influence of varying operating conditions on O_{shoot}; (b) the influence of varying operating conditions on R_{time}.

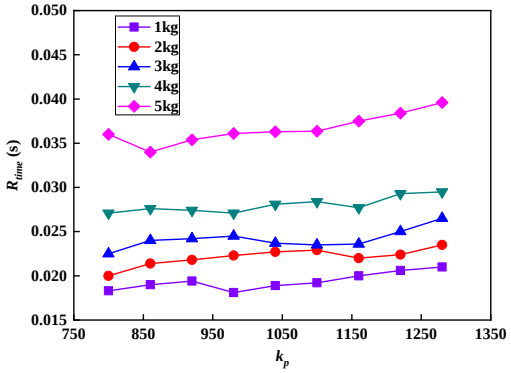
The control parameters were set to k_p = 1040 and k_i = 103, while the responsiveness performance of the joint module was tested under different conditions of W_l and V_c. The test results are depicted in Fig. 3.

Based on the illustrations in Fig. 3, as W_l increases, both O_{shoot} and R_{time} of the joint module exhibit a gradual increasing trend. This is attributed to the increased inertia and inertial forces, caused by a larger load, making it more challenging for the joint module to respond quickly and reach a steady state. Similarly, an increase in V_c, O_{shoot} and R_{time} also demonstrates an increasing trend. This is due to the higher target velocity, which requires the joint module to exhibit longer acceleration time. Moreover, the increase in velocity leads to an increase in inertial forces, resulting in larger values of O_{shoot} and R_{time}.

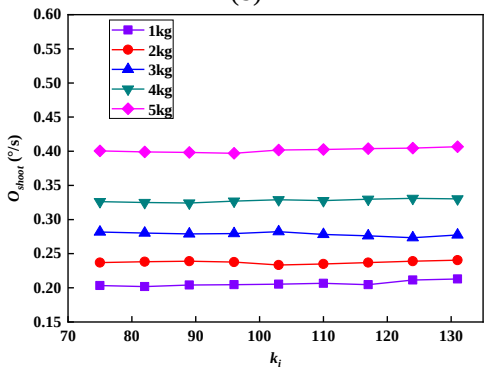
Next, the influence of k_p on the responsiveness performance of the joint module was obtained, considering V_c = 7.3 °/s and k_i = 103, as depicted in Fig. 4a,b. Following, at V_c = 7.3 °/s and k_p = 1040, the influence of k_i on the responsiveness performance of the joint module was derived, as illustrated in Fig. 4c,d.



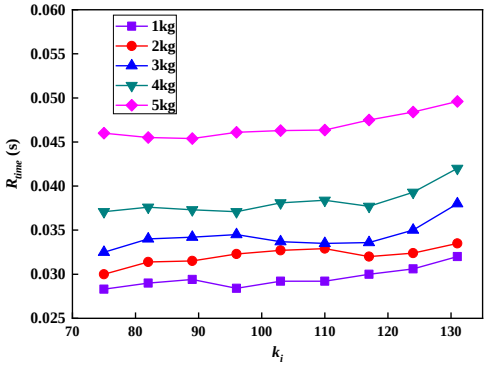
(a)



(b)



(c)



(d)

Fig. 4. The influence of control parameter variations on the responsiveness performance of the joint module: (a) the influence of k_p on O_{shoot}; (b) the influence of k_p on R_{time}; (c) the influence of k_i on O_{shoot}; (d) the influence of k_i on R_{time}.

According to Fig. 4a, as k_p increases, O_{shoot} initially decreases and then increases, indicating that within a certain range, an optimal increase in k_p leads to a more responsive system, reduces the overshoot and enhances stability and accuracy. However, when k_p exceeds a certain value, it leads to excessive response and causes overshoot rebound. Based on Fig. 4c, as k_i increases, O_{shoot} slowly increases.

In Fig. 4b,d, it can be observed that, as k_p or k_i increases, R_{time} demonstrates an increasing trend. This is attributed to the heightened sensitivity and instability of the system at higher values of k_p or k_i. In

order to maintain system stability, additional adjustments and balancing processes are required, resulting in an extended response time.

It is important to note that, both operational conditions and control parameters exhibit clear nonlinear characteristics in their impact on the responsiveness performance of the joint module. This suggests that, the responsiveness performance is influenced by multiple factors and cannot be simply described by linear relationships. A MA algorithm optimized BPNN model, also known as the MA-BPNN model, is used to achieve more accurate predictions of the responsiveness performance. This model excels in nonlinear modeling and adaptability, effectively handling the complex factors and relationships involved in the responsiveness performance. By training the model and adjusting its weights and thresholds, it can learn and capture the nonlinear relationship between inputs and outputs, thereby facilitating predictions of the responsiveness performance of the joint module.

Moreover, in order to obtain a larger and more scientifically sound sample data on the responsiveness performance of the joint module, orthogonal experiments, designed using the quasi-level method, were conducted. Based on the experiments, information about the responsiveness performance of the joint module, under different values of factors, were obtained, as listed in Table 2.

Table 2. The results of the orthogonal experiments on the responsiveness performance of the joint module.

Number	M_l (Kg)	V_c (°/s)	k_p	k_i	O_{shoot} (°/s)	R_{time} (s)
1	1	4.5	1280	131	0.38	0.0335
2	1	9.9	1220	89	0.92	0.0335
3	1	5.4	920	131	0.30	0.0260
4	1	8.1	800	110	0.44	0.0190
5	1	10.8	1280	96	1.91	0.0135
6	1	6.3	860	117	0.37	0.0420
7	1	9.0	1040	89	0.40	0.0220
8	1	3.6	1220	103	0.19	0.0380
9	1	9.9	1100	96	0.42	0.0225
10	1	10.8	980	124	0.73	0.0115
11	1	7.2	1040	117	0.36	0.0180
12	1	6.3	980	110	0.12	0.0290
13	1	3.6	800	75	0.21	0.0595
14	1	4.5	860	103	0.23	0.0410
15	1	9.0	1160	124	0.68	0.0130
16	1	8.1	1100	82	0.42	0.0215
17	1	7.2	920	82	0.12	0.0360
18	1	5.4	1160	75	0.33	0.0405
19	2	10.8	1220	75	0.48	0.0170
…	…	…	…	…	…	…
81	5	4.5	1160	89	0.27	0.0295

A total of 81 sample groups, as listed in Table 2, were randomly selected, while 75 groups were designated as the training set for the establishment of the MA-BPNN model in the following analysis. The remaining 6 groups were assigned as the validation set for evaluating and validating the model.

PREDICTION OF RESPONSIVENESS PERFORMANCE OF THE JOINT MODULE BASED ON THE MA-BPNN MODEL

This study has developed a MA-BPNN model for predicting the responsiveness performance of the joint module that can overcome the limitations and drawbacks of traditional BPNN models. The proposed model integrates the powerful nonlinear mapping capability and robust adaptability of the BPNN model, as well as the global search and adaptive adjustment abilities of the MA algorithm. It can avoid local optimal issues and improve the efficiency of the model in finding global optimal solutions.

The Optimization Process of MA Algorithm

It is assumed that mayflies exist in an *n*-dimensional space, while their positions and velocities can be respectively represented by an *n*-dimensional vector. The position vector is denoted as $x=\left(x_1,x_2,\cdots,x_n\right)^{\mathrm{T}}$ and the velocity vector is denoted as $v=\left(v_1,v_2,\cdots,v_d\right)^{\mathrm{T}}$. These position and velocity vectors can be adjusted and updated to perform a search and optimization loop.

Assuming a population of mayflies comprises *N* male and female individuals. According to the *t*-th iteration, the position, velocity and historical best position of the *i*-th male mayfly in the *j*-th dimensional search space are denoted as x_{ij}^t, v_{mij}^t and $P_{best_{ij}}$ respectively. The global best position of the population in the *j*-th dimensional search space is represented by G_{best_j}. Then, the male mayflies update their current velocities and positions based on Eqs. (1) and (2), in order to optimize their movement within the search space.

$$v_{mij}^{t+1}=\begin{cases}\zeta v_{mij}^t+a_1\mathrm{e}^{-\beta r_P^2}\left(P_{best_{ij}}-x_{ij}^t\right)+a_2\mathrm{e}^{-\beta r_G^2}\left(G_{best_j}-x_{ij}^t\right),\\ \text{if } F_3(x_{ij}^t)>F_3\left(G_{best_j}\right)\\ \zeta v_{mij}^t+da\cdot r_c,\\ \text{if } F_3(x_{ij}^t)\le F_3\left(G_{best_j}\right)\end{cases} \tag{1}$$

$$x_{ij}^{t+1}=x_{ij}^t+v_{mij}^{t+1} \tag{2}$$

where, ζ is the dynamic inertia coefficient; a_1 and a_2 are the attraction coefficients for male mayflies; β is the visibility coefficient; r_P is the distance, which is measured in Euclidean terms, between the current

position of the male mayfly and its historical best position $P_{best_{ij}}$; r_G is the distance between the current position of the male mayfly and the global best position G_{best_j}; da is the dance coefficient used to describe the process of attracting female mayflies; r_c is a random number, $r_c \in [-1,1]$; $F_3(x)$ is the fitness function.

After the t-th iteration, the position and velocity of the i-th female mayfly, in the j-th dimensional search space, are assumed to be represented by y_{ij}^t and v_{fij}^t, respectively. Then, the female mayflies update their current velocities and positions based on Eqs. (3) and (4).

$$v_{fij}^{t+1} = \begin{cases} \zeta v_{fij}^t + a_3 e^{-\beta r_{mf}^2}\left(x_{ij}^t - y_{ij}^t\right), \\ \text{if } F_3(y_{ij}^t) > F_3(x_{ij}^t) \\ \zeta v_{fij}^t + fl \cdot r_c, \\ \text{if } F_3(y_{ij}^t) \le F_3(x_{ij}^t) \end{cases} \tag{3}$$

$$y_{ij}^{t+1} = y_{ij}^t + v_{fij}^{t+1} \tag{4}$$

where, a_3 is the attraction coefficient for female mayflies; r_{mf} is the distance between female and male mayflies; fl is the random flight coefficient, used to describe the random flight path of female mayflies, when they are not attracted by male mayflies.

The mating process of mayflies involves randomly selecting a subset of individuals from the male and female populations for pairing. Superior males mate with superior females, while suboptimal males mate with the other remaining females. The offsprings generated using Eqs. (5) and (6) represent the progeny resulting from the mating process.

$$offspring_1 = r_l \cdot male + (1 - r_l) \cdot female \tag{5}$$

$$offspring_2 = r_l \cdot female + (1 - r_l) \cdot male \tag{6}$$

where, $offspring_1$ and $offspring_2$ represent the two offspring mayflies; r_l is a random number, $r_l \in [-1,1]$; *male* and *female* respectively represent the male and female mayflies.

Structure Determination of the BPNN Model

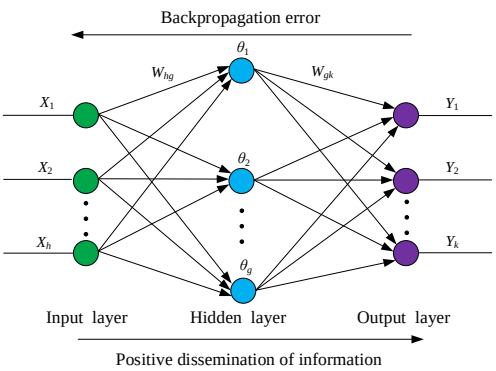


Fig. 5. The topological structure of BPNN model.

The topological structure of the BPNN model is illustrated in Fig. 5. $X_1, X_2, \ldots, X_h$ represent the input layer neurons, $\theta_1, \theta_2, \ldots, \theta_g$ represent the hidden layer neurons, while $Y_1, Y_2, \ldots, Y_k$ represent the output layer neurons. W_{hg} represents the weight connection between the h-th neuron in the input layer and the g-th neuron in the hidden layer, while W_{gk} represents the weight connection between the g-th neuron in the hidden layer and the k-th neuron in the output layer.

In this study, the BPNN model has input neurons represented by W_l, V_c, K_p and K_i, and output neurons represented by O_{shoot} and R_{time}. The input layer consists of four nodes (n_{in} = 4), while the output layer consists of two nodes (n_{out} = 2). The number of nodes in the hidden layer, denoted as n_{hide}, is determined by Eq. (7).

$$n_{hide} = \text{int}\left(\sqrt{n_{in} + n_{out}} + a_c\right) \tag{7}$$

where a_c is a constant, $a_c \in [1,10]$.

The training process of the BPNN model is as follows:

Step 1: Initialization. The initialization of input layer neuron weights is performed according to normal distribution with mean 0 and standard deviation 1. The thresholds of the hidden and output layer neurons are set as random real numbers within the range [0, 0.001].

Step 2: Forward propagation. The training set samples are introduced to the input layer of the BPNN model. The data are transmitted to the hidden layer neurons considering weights and thresholds. The activation function, written as $F_1(x) = \left(e^x - e^{-x}\right) / \left(e^x + e^{-x}\right)$, is applied to the hidden layer neurons, while the g-th neuron (θ_g) is calculated as:

$$\theta_g = F_1\left(\sum_{g=1}^{n_{hide}} W_{hg} X_h - T_g\right) \tag{8}$$

where, T_g represents the threshold value for the g-th neuron in the hidden layer.

The output of the hidden layer is then transmitted to the output layer neurons taking into consideration weights and thresholds. The activation function, written as $F_2(x) = e^x / \left(e^x + e^{-x}\right)$, is applied to the output layer neurons, while the k-th neuron (Y_k) is calculated as:

$$Y_k = F_2\left(\sum_{k=1}^{n_{out}} W_{gk} \theta_g - T_k\right) \tag{9}$$

where, T_k represents the threshold value for the k-th neuron in the output layer.

Step 3: Prediction error calculation. The prediction error (E_u) is computed by comparing the predicted output (denoted as y_{pt}) of the output layer neuron to the expected output (denoted as y_e) in the training set of samples. It can be expressed as

follows:

$$E_u = \left| y_{pt} - y_e \right| \tag{10}$$

Step 4: Backpropagation. The backpropagation algorithm is used to update and adjust the weights and thresholds of each neuron, aiming to minimize the E_u of the BPNN model.

Step 5: Training iterations. The weights and thresholds are adjusted through the iteration of steps 2, 3 and 4, until the convention E_u < 0.0001 is achieved, resulting in the output of prediction results.

During the training process of the BPNN model, the mean-square-error (MSE) values of E_u, as obtained based on different values of n_{hide}, are listed in Table 3. MSE(E_u) is defined as a dimensionless quantity, used to evaluate the performance of the model. According to Table 3, the minimum value of MSE(E_u), which is 0.026, is achieved when n_{hide} = 9, indicating that the BPNN model performs best, in terms of prediction, under this configuration. Therefore, it is decided to set the number of nodes in the input layer, hidden layer and output layer of the BPNN model as 4, 9 and 2, respectively.

Table 3. The MSE of E_u obtained under different values of n_{hide}.

n_{hide}	MSE(E_u)	n_{hide}	MSE(E_u)
3	0.046	8	0.039
4	0.037	9	0.026
5	0.055	10	0.039
6	0.037	11	0.032
7	0.038	12	0.030

Establishment of the MA-BPNN Model

The iterative process of the MA-BPNN model is illustrated in Fig. 6 and can be described in detail as follows:

Step 1: Initialization. The training set samples are imported. The parameters for the MA algorithm are set as follows: N = 25, β = 2, a_1 = 0.2, a_2 = 0.3 and a_3 = 0.2. The position of each mayfly corresponds to a set of weight and threshold parameters for the BP neural network, while the velocity represents the step size for updating these parameters, during the iterative process. The weights of the BP neural network are randomly initialized according to the normal distribution with mean 0 and standard deviation 1, while the initial thresholds are set as random real numbers within the range [0, 0.001]. The positions of the individuals in the population are initialized, while their initial velocities are set to 0.3. The fitness value, corresponding to the position of each individual element, is calculated using Eq. (11), whereas G_{best_j} and $F_3\left(G_{best_j}\right)$ are recorded.

$$F_3(x) = \mathrm{MSE}(E_u) \tag{11}$$

Step 2: Update of the velocities and positions of male population. The v_{mij}^{t+1} and x_{ij}^{t+1} are updated using Eqs. (1) and (2). Furthermore, G_{best_j} and $F_3\left(G_{best_j}\right)$ are updated accordingly.

Step 3: Update of the velocities and positions of female population. The v_{fij}^{t+1} and y_{ij}^{t+1} are updated using Eqs. (3) and (4). Furthermore, G_{best_j} and $F_3\left(G_{best_j}\right)$ are updated accordingly.

Step 4: Generation of offspring mayflies. The mayfly population is updated according to Eqs. (5) and (6).

Step 5: Update G_{best_j} and $F_3\left(G_{best_j}\right)$. If the target fitness value (0.015) is achieved, the iteration loop stops; otherwise, the process proceeds to Step 2 for further iterations.

Step 6: Obtain optimized weights and thresholds.

Step 7: Forward propagation of BPNN. The input layer data is transmitted to the hidden layer neurons, using the optimized weights and thresholds. The hidden layer neurons are computed based on Eq. (8). Next, the hidden layer data are transmitted to the output layer neurons, which have been computed according to Eq. (9).

Step 8: Prediction error calculation. Determination of the value of E_u.

Step 9: Backpropagation. Use of the backpropagation algorithm to adjust weights and thresholds to minimize E_u.

Step 10: Training iteration. By repeatedly executing Steps 7, 8, and 9 and adjusting weights and thresholds until E_u < 0.0001, the final output is generated.

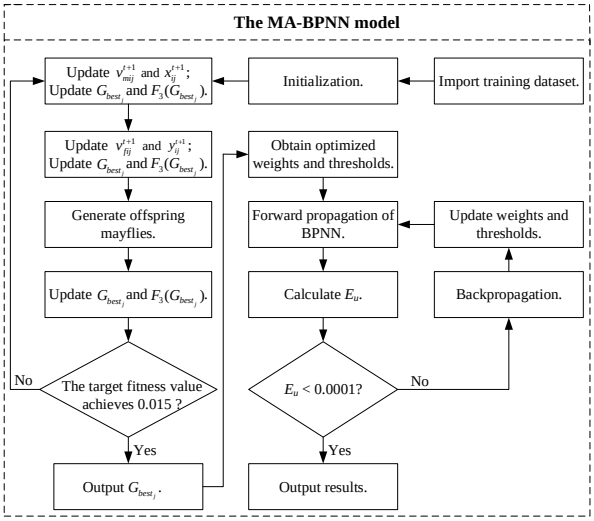


Fig. 6. The flowchart of the MA-BPNN model.

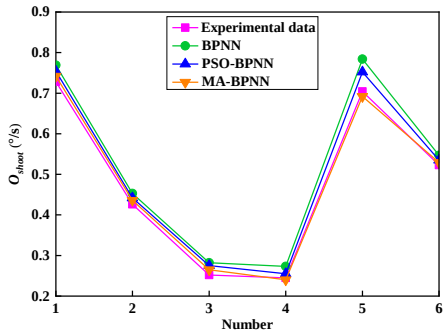
Prediction of Responsiveness Performance

Based on the validation set sample data, the

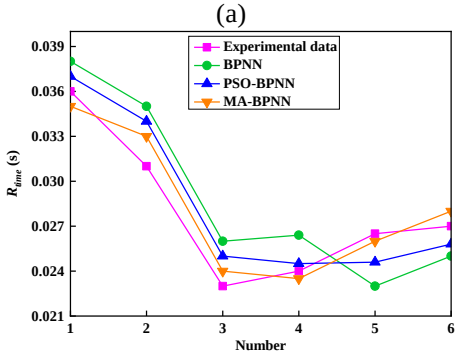
responsiveness performance of the joint module is predicted using the BPNN model, the PSO-BPNN and the MA-BPNN model. The results are presented in Table 4, while a comparative illustration is presented in Fig. 7. According to the results, the predictions of the MA-BPNN model for O_{shoot} and R_{time} are closer to the experimental values with smaller errors. In contrast, the BPNN model and the PSO-BPNN model show significant deviations from the experimental values with larger errors. Therefore, it can be concluded that the MA-BPNN model provides more accurate predictions of the responsiveness performance of the joint module.

Table 4. The prediction results of the responsiveness performance of the joint module using the BPNN, PSO-BPNN and MA-BPNN models.

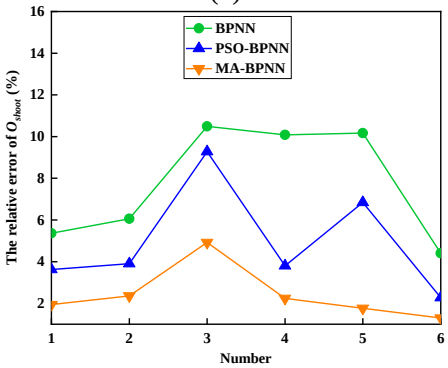
Number	W_l (Kg)	V_c (°/s)	k_p	k_i	Experimental data		BPNN		PSO-BPNN		MA-BPNN	
					O_{shoot} (°/s)	R_{time} (s)	O_{shoot} (°/s)	R_{time} (s)	O_{shoot} (°/s)	R_{time} (s)	O_{shoot} (°/s)	R_{time} (s)
1	1	10.8	1280	96	0.728	0.0360	0.769	0.0380	0.754	0.0370	0.742	0.0350
2	1	9.0	1040	89	0.426	0.0310	0.453	0.0350	0.442	0.0340	0.436	0.0330
3	2	4.5	800	82	0.252	0.0230	0.282	0.0260	0.275	0.0250	0.265	0.0240
4	3	3.6	860	96	0.245	0.0240	0.273	0.0264	0.255	0.0245	0.24	0.0235
5	4	6.3	1220	124	0.704	0.0265	0.784	0.0230	0.752	0.0246	0.692	0.0260
6	5	5.4	1220	96	0.523	0.0270	0.547	0.0250	0.535	0.0258	0.529	0.0280



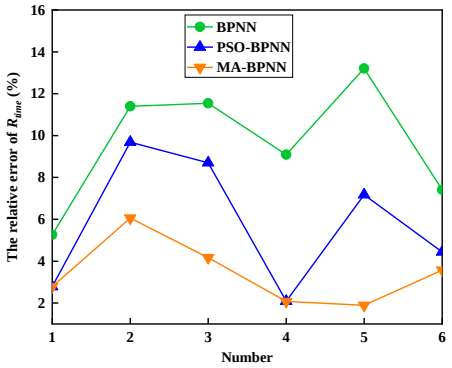
(a)



(b)



(c)



(d)

Fig. 7. The comparison of the prediction results of the responsiveness performance using the BPNN, PSO-BPNN and MA-BPNN models: (a) the comparison of O_{shoot} results; (b) the comparison of R_{time} results; (c) the comparison of relative errors for O_{shoot} results; (d) the comparison of relative errors for R_{time} results.

Furthermore, in order to evaluate the models, the coefficient of determination (R^2), the MSE of overshoot prediction (MSE(O_{shoot})), the MSE of response time prediction (MSE(R_{time})) and the model execution time (M_{time}) were used as evaluation metrics. The results are listed in Table 5.

Table 5 demonstrates that the R^2 value of the MA-BPNN model surpasses that of the other models, suggesting a superior fit and enhanced predictive accuracy. Moreover, both the MSE(O_{shoot}) and MSE(R_{time}) values, obtained by the MA-BPNN model, are lower than those of the other models, indicating smaller errors and improved performance.

Additionally, the MA-BPNN model exhibits a shorter execution time. Therefore, it can be inferred that the MA-BPNN model performs better in terms of predictive accuracy, error control and computational efficiency.

Table 5. The comparison of the evaluation metrics for the BPNN, PSO-BPNN and MA-BPNN models.

Model	Evaluation indicators			
	R^2	MSE(O_{shoot}) (°/s)^2	MSE(R_{time}) (s^2)	M_{time} (s)
BPNN	0.973	0.0254	0.0266	0.38254
PSO-BPNN	0.982	0.0186	0.0193	0.10142
MA-BPNN	0.991	0.0138	0.0135	0.05261

OPTIMIZATION OF CONTROL PARAMETERS FOR THE JOINT MODULE BASED ON THE MA-BPNN MODEL

Based on the Latin hypercube sampling method and the established MA-BPNN model, an optimization of the control parameters of the joint module, under common operating conditions, will be implemented.

Sampling of Control Parameters

Latin hypercube sampling is a method used to achieve a uniform distribution and randomness of parameter points, within a finite number of samples, aiming to improve sampling efficiency and obtain a representative and reliable sample set. In this study, the Latin hypercube sampling method is adopted, to sample the values of control parameters (k_p and k_i) for the joint module. The sampling process is as follows:

(1) Determination of parameter ranges and sample point quantity.

The range of k_p was set to [800, 1290] and for the k_i values, it was set to [75, 131]. The number of sample points, n_s, was set to be 200.

(2) Construction of Latin hypercube matrix $\boldsymbol{L}$.

$\boldsymbol{L}$ has a size of 200×2. For the f-th column of $\boldsymbol{L}$, generate a random permutation $\boldsymbol{P}_f$ within the range [0, 1, 2, ..., n_s-1]. For the q-th row of $\boldsymbol{L}$, the following equation was applied:

$$\boldsymbol{L}[q, f] = \left(\boldsymbol{P}_f[q] + \text{random}(0,1)\right) \cdot \left(b_f - a_f\right) / n_s + a_f \quad (12)$$

where, a_f and b_f are the lower and upper bounds of the range for control parameters.

The resulting distribution of the Latin hypercube sampling points is shown in Fig. 8.

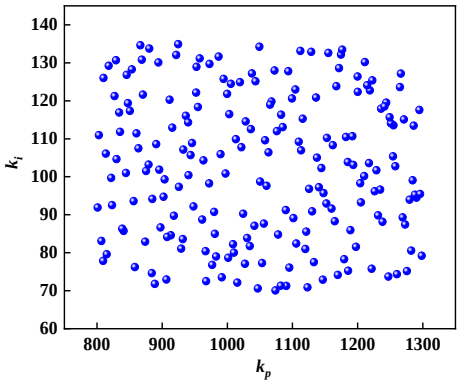


Fig. 8. The distribution of Latin hypercube sampling points.

Optimization of Control Parameters

In order to enhance the comparability and interpretability of the data, normalization was applied to the control parameters data of k_p and k_i in Fig. 8, mapping them to the range [0,1]. In this study, the goal of control parameter optimization is to achieve the optimal state of the joint module system, under specific operating conditions. In order to quantitatively evaluate the response performance of the joint module, a respective indicator K is defined, according to the following equation:

$$K = \delta_O \cdot \left|O_{shoot}\right| + \delta_R \cdot R_{time} \quad (13)$$

where, δ_O and δ_R are the weighting coefficients for O_{shoot} and R_{time}, respectively.

Based on the analysis in Fig. 4 and the emphasis on system stability and fast response, it is set δ_O = 0.6 with a higher weight to emphasize reducing overshoot, and δ_R = 0.4 with a lower weight to prioritize fast response.

In this study, the load (W_l) of the joint module is set to values of 1 kg, 2 kg, 3 kg, 4 kg, and 5 kg, while the velocity (V_c) is set to 3.6 °/s, 5.4 °/s, 7.2 °/s, 9 °/s, and 10.8 °/s, resulting in a total of 25 common operating conditions.

For each operating condition, the control parameter data of k_p and k_i serve as input to the MA-BP model, which provides the responsiveness performance indicators O_{shoot} and R_{time} of the joint module. By minimizing the value of K, the Pareto front solutions for the control parameters values for each operating condition can be obtained. Indicatively, Fig. 9 shows the Pareto front solutions of the control parameters for the operating condition of the joint module where W_l = 1 kg and V_c = 3.6 °/s.

In this study, the control parameter combinations with the minimum k_p values are selected as the Pareto optimal solutions for each operating condition. The decision to retain the control parameter combinations with the minimum k_p values was motivated by their ability to effectively balance factors, such as overshoot, excessive system response and response time, thereby achieving superior

performance. Ultimately, the Pareto optimal solutions for the control parameters of the joint module, under common operating conditions, are obtained, as listed in Table 6.

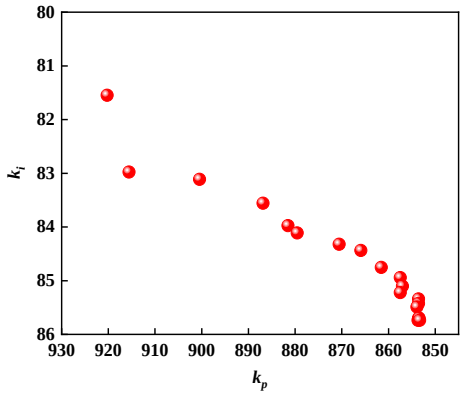


Fig. 9. The Pareto front solutions of the control parameters for the joint module at W_l = 1 kg and V_c = 3.6 °/s.

Table 6. The Pareto optimal solutions for control parameters of the joint module under common operating conditions.

Number	W_l (Kg)	V_c (°/s)	K_p	k_i
1	1	3.6	852	85
2	1	5.4	869	110
3	1	7.2	874	121
4	1	9.0	888	88
5	1	10.8	933	115
6	2	3.6	865	111
7	2	5.4	875	98
8	2	7.2	897	118
9	2	9.0	902	102
10	2	10.8	941	85
11	3	3.6	875	106
12	3	5.4	884	105
13	3	7.2	911	81
14	3	9.0	925	92
15	3	10.8	1013	89
16	4	3.6	941	110
17	4	5.4	961	84
18	4	7.2	1024	71
19	4	9.0	1046	77
20	4	10.8	1106	79
21	5	3.6	1056	98
22	5	5.4	1058	70
23	5	7.2	1146	112
24	5	9.0	1159	94
25	5	10.8	1211	78

Furthermore, based on Table 6, the variation trends of k_p and k_i, in the Pareto optimal solutions set for the control parameters of the joint module, under common operating conditions, are obtained and illustrated in Fig. 10.

Based on Fig. 10a, k_p increases along the increase of W_l and V_c. However, an increase in V_c leads to greater fluctuations in k_p. According to Fig. 10b, k_i decreases as W_l, increases and it increases with the increase of V_c. Overall, k_p is more sensitive to operating conditions. This implies that, in designing control systems, it is necessary to consider appropriately increasing the proportional control parameter k_p to meet the requirements of the system, when facing heavier loads and higher joint velocities.

Additionally, the Pareto optimal solutions for the control parameters, under common operating conditions, provide designers with a range to guide them, allowing them to select control parameters based on specific requirements and achieve optimal control performance and stability.

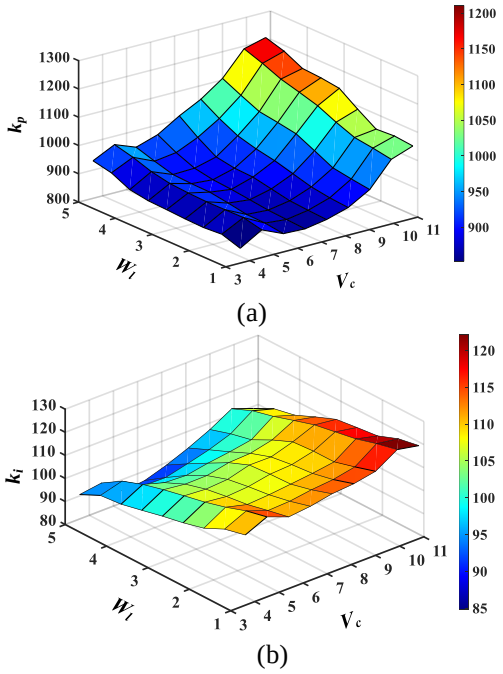


Fig. 10. The variation trends of k_p and k_i in the Pareto optimal solutions of control parameters under common operating conditions: (a) the trend of k_p variation; (b) the trend of k_i variation.

CONCLUSIONS

This study proposes a method for predicting the responsiveness performance of joint modules and an optimization approach for their control parameters, based on the MA-BPNN model. The following conclusions can be derived:

(1) The responsiveness performance of the developed joint module was tested, revealing that as the load or angular velocity increased, there was a corresponding increase in overshoot and response time. Similarly, an increase in the proportional or integral gain coefficient led to longer response times. The proportional gain coefficient caused an initial decrease in overshoot, followed by an increase, while

the integral gain coefficient resulted in a gradual increase in overshoot. Notably, the joint module's responsiveness was influenced by both operating conditions and control parameters, indicating its complex and multifactorial character that cannot be easily described by linear relationships.

(2) A predictive MA-BP model was established to assess the responsiveness performance of the joint module. The comparison of the the predictions generated by the MA-BPNN model, the PSO-BPNN model and the BPNN model, concluded that, the MA-BPNN model produced more accurate predictions with smaller errors, in terms of overshoot and response time. Furthermore, the MA-BPNN model exhibited superior fitting performance and a shorter runtime. Therefore, the superiority of the MA-BPNN model, in accuracy and efficiency, when predicting the responsiveness performance of the joint module, is evident.

(3) This study used Latin hypercube sampling and the MA-BP model, to obtain Pareto optimal solutions for control parameters of the joint module, under common operating conditions. The results provide designers with guidelines to select control parameters for optimal control performance and stability. It is observed that, the proportional gain coefficient is more sensitive to variations, in operating conditions within these Pareto optimal solutions. Thus, suitable increase of the proportional gain coefficient is necessary, when designing control systems, in order to meet system requirements with heavier loads and faster joint speeds.

ACKNOWLEDGMENT

The authors gratefully acknowledge the National Natural Science Foundation of China coded 52305061 and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China coded 23KJB460011.

REFERENCES

Zhao, Y.L., Huang, H.P., Chiang, P.C., Chen, J.H., Chen Y.C., Wang, H.T., "Development of Robots for People with Dementia: Technologies and Applications," *Journal of the Chinese Society of Mechanical Engineers*, Vol. 44, No.4, pp. 351-360 (2023).

Zhen, S.C., Cui, W.X., Liu, X.L., Meng, G.J., Chen, Y.H., "A novel model-based robust control design for collaborative robot joint module," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 236, No. 9, pp. 4520-4532 (2022).

Cai, L.G., Hu, Q.S., Liu, Z.F., Yang, C.B., Zhang, T., "Stress calculation and fatigue life evaluation on cup-type flexspline under bending and torsion in harmonic drive," *Journal of the Chinese Society of Mechanical Engineers*, Vol. 40, No. 5, pp. 471-480 (2019).

Hu, Q.S., Liu, Z.F., Yang, C.B., Xie, F.G., "Research on dynamic transmission error of harmonic drive with uncertain parameters by an interval method," *Precision Engineering*, Vol. 68, pp. 285-300 (2021).

Huang, K., Xian, Y.J., Zhen, S.C., Sun, H., "Robust control design for a planar humanoid robot arm with high strength composite gear and experimental validation," *Mechanical Systems and Signal Processing*, Vol. 155, pp. 107442-1-12 (2021).

Tian, W.J., Xu, M., Zhang, X.P., Guo, X., Wang, L.N., Huang, T., "Repeatability prediction of 6-DOF hybrid robot based on equivalent error model of actuated joint," *Measurement*, Vol. 207, pp. 112377-1-15 (2023).

Cui, G.Y., Li, B., Tian, W., Liao, W.H., Zhao, W., "Dynamic modeling and vibration prediction of an industrial robot in manufacturing," *Applied Math Model*, Vol. 105, pp. 114-136 (2022).

Bittencourt, A.C. and Axelsson, P., "Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data," *IEEE/ASME Transactions on Mechatronics*, Vol. 19, No. 5, pp. 1694-1706 (2014).

Jiang, P., Wang, Z.X., Li, X.B., Wang, X.V., Yang, B.D., Zheng, J.J., "Energy consumption prediction and optimization of industrial robots based on LSTM," *Journal of Manufacturing Systems*, Vol. 70, pp. 137-148 (2023).

Nohooji, H.R., Zaraki, A., Voos, H., "Actor–critic learning based PID control for robotic manipulators," *Applied Soft Computing*, Vol. 151, pp. 111153-1-29 (2024).

Zhang, X., Zhou, H., Liu, J.G., Ju, Z.J., Leng, Y.Q., Yang, C.G., "A practical PID variable stiffness control and its enhancement for compliant force-tracking interactions with unknown environments," *Science China Technological Sciences*, Vol. 66, No. 10, pp. 2882-2896 (2023).

Ang, K.H., Chong, G., Li, Y., "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 4, pp. 559-576 (2005).

Angel, L. and Viola, J., "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, Vol. 79, pp. 172-188 (2018).

Zhong, J., Hu, S.G., Wang, Z.C., Han, Z.F., "A cascade BP neural network tuned PID controller for a high-voltage cable-stripping

robot," *Micromachines*, Vol. 14, No. 3, pp. 689-1-12 (2023).

Nikdel, N., Badamchizadeh, M.A., Azimirad, V., Nazari, M.A., "Adaptive backstepping control for an n-degree of freedom robotic manipulator based on combined state augmentation," *Robotics and Computer Integrated Manufacturing*, Vol. 44, pp. 129-143 (2017).

Baek, J., Jin, M., Han, S., "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, Vol. 63, No. 6, pp. 3628-3637 (2016).

Méndez, G.M., Montes Dorantes, P.N., Aracelia, M.A., "Dynamic adaptation of the PID's gains via Interval type-1 non-singleton type-2 fuzzy logic systems whose parameters are adapted using the backpropagation learning algorithm," *Soft Computing*, Vol. 24, No. 2, pp. 17-40 (2020).

Han, J.L., Shan, X.L., Liu, H.T., Xiao, J.L., Huang, T., "Fuzzy gain scheduling PID control of a hybrid robot based on dynamic characteristics," *Mechanism and Machine Theory*, Vol. 184, pp. 105283-1-16 (2023).

Van, M., Do, X.P., Mavrovouniotis, M., "Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *ISA Transactions*, Vol. 96, pp. 60-68 (2020).

Liu, E.J., Yang, Y.N., Yan, Y., "Spacecraft attitude tracking for space debris removal using adaptive fuzzy sliding mode control," *Aerospace Science and Technology*, Vol. 107, pp. 106310-1-13 (2020).

Carlucho, I., Paula, M.D., Acosta, G.G., "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," *ISA Transactions*, Vol. 102, pp. 280-294 (2020).

Son, N.N., Kien, C.V., Anh, H.P.H., "A novel adaptive feed-forward-PID controller of a SCARA parallel robot using pneumatic artificial muscle actuator based on neural network and modified differential evolution algorithm," *Robotics and Autonomous Systems*, Vol. 96, pp. 65-80 (2017).

Zheng, K.M., Zhang, Q.J., Peng, L., Zeng, S.S., "Adaptive memetic differential evolution-back propagation-fuzzy neural network algorithm for robot control," *Information Sciences*, Vol. 637, pp. 118940-1-19 (2023).

Liang, T.L., Glossner, J., Wang, L., Shi, S.B., Zhang, X.T., "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, Vol. 461, pp. 370-403 (2021).

Jian, B.L., Wang, C.K., Hsieh, C.T., Kuo, Y.P., Houng, M.C., Yau, H.T., "Predicting spindle displacement caused by heat using the general regression neural network," *The International Journal of Advanced Manufacturing Technology*, Vol. 104, pp. 4665-4674 (2019).

Wang, C.C., Kuo, P.L., Chen, G.Y., "Machine learning prediction of turning precision using optimized XGBoost model," *Applied Sciences*, Vol, 12, pp. 7739-1-19 (2022).

Rahideh, A., Shaheed, H.M., "Neural network-based modelling of a two-degrees-of-freedom twin rotor multiple input, multiple output system using conjugate gradient learning algorithms," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 222, No. 6, pp. 757-771, (2008).

Wang, Wei., Pu, Y.M., Li, W., "A parallel Levenberg-Marquardt algorithm for recursive neural network in a robot control system," *International Journal of Cognitive Informatics and Natural Intelligence*, Vol. 12, No. 2, pp. 32-47 (2018).

Zuo, G.Y., Qiu, Y.K., Liu, Y.L., "Sensorless external force detection method for robot arm based on error compensation using BP neural network," *International Journal of Humanoid Robotics*, Vol. 16, No. 5, pp. 1950024-1-22 (2019).

Lin, C.J., Chu, W.L., Wang, C.C., Chen, C.K., Chen, I.T., "Diagnosis of ball-bearing faults using support vector machine based on the artificial fish-swarm algorithm," *Journal of Low Frequency Noise, Vibration and Active Control*, Vol. 39, No. 4, pp. 954-967 (2020).

Ning, Y., Jin, Y.P., Peng, Y.D., Yan, J., "Small obstacle size prediction based on a GA-BP neural network," *Applied Optics*, Vol. 61, No. 1, pp. 177-187 (2022).

Bai, B., Zhang, J.Y., Wu, X., Zhu, G.W., Li, X.Y., "Reliability prediction-based improved dynamic weight particle swarm optimization and back propagation neural network in engineering systems," *Expert Systems with Applications*, Vol. 177, pp. 114952-1-13 (2021).

Shirzadeh, M., Amirkhani, A., Tork, N., Taghavifar, Hamid., "Trajectory tracking of a quadrotor using a robust adaptive type-2 fuzzy neural controller optimized by cuckoo algorithm," *ISA Transactions*, Vol. 114, pp. 171-190 (2021).

Yan, Y.L., Sun, T.S., Ren, T., Ding, L., "Enhanced grip force estimation in robotic surgery: A sparrow search algorithm-optimized backpropagation neural network approach," *Mathematical Biosciences and Engineering*, Vol. 21, No.3, pp. 3519-3539 (2024).

基於 MA-BPNN 模型的機器人一體化關節模組控製參數優化

胡秋實　　李恒　　李浩強　　王國偉
李磊
江蘇科技大學機械工程學院

摘 要

本研究提出了一種預測關節模組響應性能幷優化其控制參數的方法。通過測試關節模組在不同工況與控制參數下的超調量和響應時間，揭示了工況與控制參數對關節模組響應性能的非綫性影響。將 BPNN 模型與 MA 算法相結合，建立了 MA-BPNN 模型，該模型相較於傳統模型，具有更高的預測精度和計算效率。利用拉丁超立方采樣和 MA-BPNN 模型，確定了關節模組控制參數的 Pareto 最優解。研究指出，高負載和關節速度下，應適當提高比例增益繫數，以滿足系統需求。該研究成果對關節模塊的設計和優化具有重要應用價值。