Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/31/20241088

Current study on PID control method with optimized BP neural network based on particle swarm optimization

Han Wang

School of Robotics, Hunan university, Changsha, China

206082128@mail.sit.edu.cn

Abstract. At present, robots show great application value in various industries, and the progress of the industry constantly puts forward higher requirements for the performance of robots, and robot control is an important part of robot application. It is difficult for the traditional PID controller to implement online tuning when it is faced with actual objects. Consequently, the neural network algorithm has been included into robot control methods in recent years; the PID controller based on BP neural network is the subject of this research. This study presents the current state of research, the basic concept behind BP neural networks, how they are used in controller design, and how to optimize the PID parameters of BP neural networks using a particle swarm optimization method. To increase system stability, the neural network is coupled with a PID controller. The adaptive learning capability of the neural network is utilized to modify PID control parameters online in real time. Targeting the trouble that it slips into a local minimum easily during the BP-PID self-learning process and the refined PSO algorithm is used to improve it. It makes sure that the BP-PID system converges to the global optimal solution. The proposed method can effectively improve the system control accuracy and control stability.

Keywords: Neutral network, PID control algorithm, particle swarm optimization

1. Introduction

In today's society, neural network technology is rapidly advancing and demonstrating significant potential in various fields. Among these, robot control stands as a crucial application domain of neural networks, involving the integration of intelligence and autonomy into robotic systems to achieve more efficient, flexible, and intelligent taskionic learning model, neural networks simulate the connections and information transmission between neurons in the human brain to enhance the learning and adaptation capabilities of robot controllers robot controllers heavily rely on preprogrammed rules and fixed logic, which restricts their performance in complex and variable environments. However, the emergence of neural networks has revolutionized this situation by enabling autonomous learning. Neural networks can automatically extract nonlinear rules and patterns from data while making corresponding adjustments and decisions based on environmental changes. This flexibility and adaptability make them ideal for controlling robots. The purpose of this paper is to introduce the application of neural networks in designing robot controllers while showcasing their potential for enhancing the intelligence and autonomy of robots. It is hoped that through this research contribution, it will further advance robotics development as well as promote practical scenarios' utilization of artificial intelligence technology.

Traditional PID control method requires manual parameter tuning, which can be difficult and time-consuming for complex nonlinear systems and varying environmental conditions. BP neural network has advantages such as strong adaptability and ability to handle complex information, and can handle nonlinear relationships. Especially for coupled systems, BP neural network's adaptability is particularly prominent. The BP-PID control method has a wide range of applications in robot control. Compared with traditional PID control methods, which require manual parameter tuning, BP neural network-based PID control method can automatically obtain optimal parameter values through training without being affected by human factors, making it more suitable for complex and dynamic control tasks. So, PID control method based on BP neural network has a wide application prospect in robot control.

The setting of the initial parameters of the BP-PID control model is important for control performance. With the wrong initial value, the system is easy to fall into the local extreme state, which leads to the control instability [1]. So, a Particle Swarm Optimization (PSO) optimized BP neural network PID control method is proposed. In this method, inertia weight is mostly introduced to update the particle velocity, and linear decreasing is mostly used for optimization when adjusting [2], while in a different method also based on PSO, random inertia weight is used to update and adjust the particle velocity. Compared with the linearly decreasing inertia weight, this method not only ensures that this method has enough local search ability in the early iteration, but also ensures its global search ability in the later iteration, which can significantly improve the convergence speed and global convergence.

2. PID control

In engineering and robotics, the PID controller is a popular industrial control system that uses a common control loop feedback mechanism. It computes the error signal that exists between a system's intended and actual outputs, and then uses derivative, integral, and proportional terms to modify the control signal in order to minimize this error. The process and structure are shown in Figure 1.

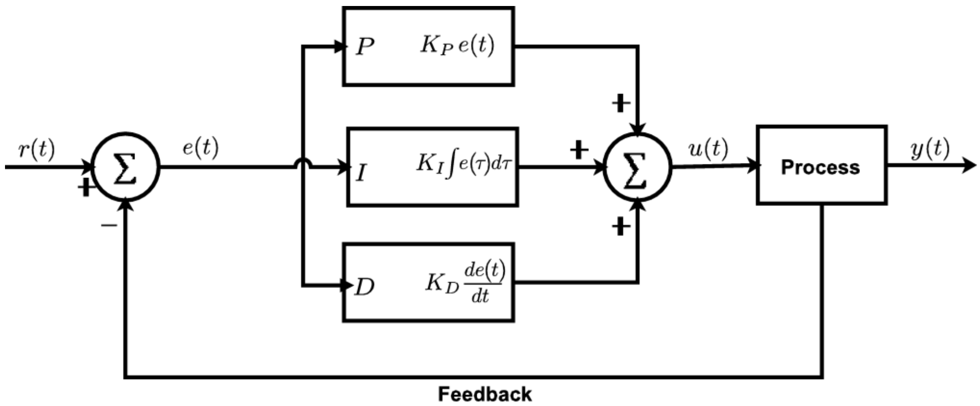

Figure 1. Standard configuration of a PID control system [3]

The output of the proportional term is directly proportional to the current error signal, whereas the integral term integrates the error signal over time and offers a corrective measure for sustained error. The derivative term provides a corrective measure based on the error signal's rate of change. The combination of these three terms enables PID control to efficiently regulate a system's output and ensure stability, even in the face of environmental changes or disturbances:

$$\Delta u(k) = u(k-1) + \Delta u(k) \tag{1}$$

$$\Delta u(k) = K_P\big(e(k) - e(k-1)\big) + K_I e(k) + K_D\big(e(k) + e(k-2) - 2e(k-1)\big) \tag{2}$$

$\Delta u(k)$is the increment and $e(k)$ is error. It can be seen from the above equation that incremental PID must find out a set of parameters through a certain method to achieve the best control performance in order to obtain the best control effect. However, when the traditional tuning method is faced with complex objects, the performance in the control system often cannot meet the demand, and there are problems such as large overshoot and long adjustment time. Therefore, a parameter tuning method with strong adaptability and learning ability in complex and uncertain environment is needed. And the BP neural network has these advantages, it is applied to the PID control, so that the control system has a better ability to learn and adapt.

3. BP neutral network

3.1. Structure of BP neutral network

The benefit of the BP neural network, which possesses nonlinear mapping features, is that it can learn on its own and adjust dynamically during training. In order for the PID motion control algorithm to attain the ideal nonlinear control mode, the network is employed to enhance and optimize the PID control algorithm [4].

Feedforward and backpropagation are the two basic phases in training the BP neural network. The feedforward stage involves computing the network's output by propagating input data layer by layer through the network. After that, the computed and desired outputs are compared, and an error value is determined. In the backpropagation step, the error is propagated backward through the network, allowing the weights and biases to be updated based on their contributions to the error.

In this paper, the 4-5-3 three-layer BP neural network structure is used as shown in Figure 2, where j, i and l correspond to neurons in the input layer, neurons in the middle layer and neurons in the output layer, respectively. The four variables of the system's output value are the input layer's neurons, the expected value, the control quantity, and the deviation. The three parameters of the controller k_P, k_I, and k_D are in the neurons of the output layer and the number of neurons in the middle layer is set to 5 in reference [7].

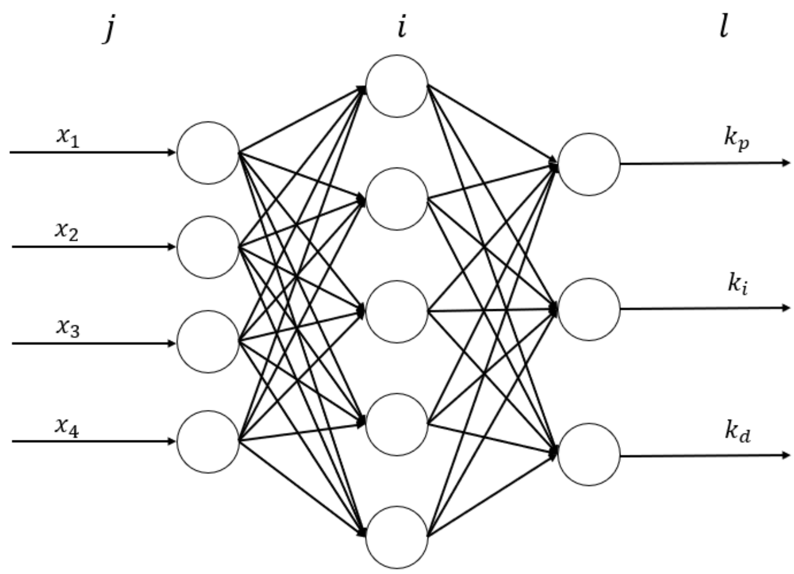

Figure 2. Structure of BP neutral network [6]

3.1.1. BP-PID controller. The BP-PID control structure shown in Figure 3 is composed of the various components of a system, such as the control algorithm and the neural network. The idea is to add the learning capabilities of the network to the control process. In the current state of the system, the three parameters of the control system are adjusted in real time. The closed-loop method is then used to implement the system.

DOI: 10.54254/2753-8818/31/20241088

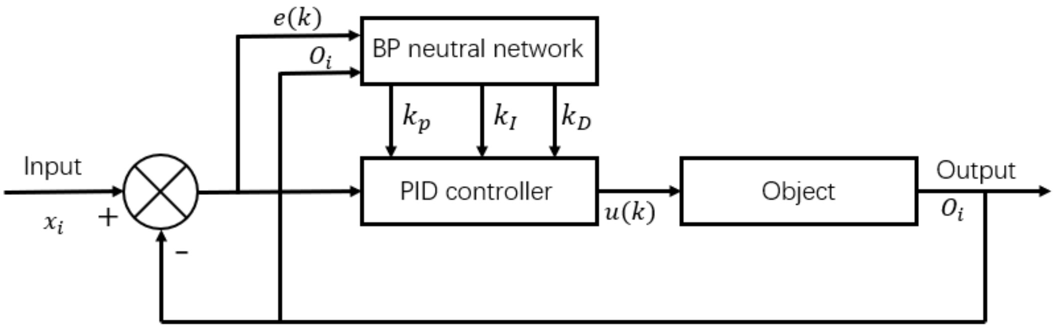

Figure 3. Structure of PID control system including BP neutral network (Picture credit: Original)

In the Figure 3, the input vetor is $x = (x_1, x_2, x_3, x_4)^T$. Sigmoid function $f(\cdot)$ can be used to process the input x_i to get the input of the input of the middle neurons $net_i^{(2)}(k)$ and the output of the middle neurons $O_i^{(2)}(k)$ as follows:

$$\begin{cases} net_i^{(2)}(k) = \sum_{j=1}^{M} w_{ij}^{(2)} x_j^{(1)} \\ O_i^{(2)}(k) = f\left(net_i^{(2)}(k)\right), i = 1,2,3,4,5 \end{cases} \tag{3}$$

$w_{ij}^{(2)}$ is the weight coefficient connecting the input layer and the middle layer. The function $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. So, the output of the network $O_i^{(3)}$ can be further obtained:

$$\begin{cases} net_i^{(3)}(k) = \sum_{j=1}^{P} w_{ij}^{(3)} O_j^{(2)} \\ O_1^{(3)}(k) = k_P(k) = h\left(net_1^{(3)}(k)\right) \\ O_2^{(3)}(k) = k_I(k) = h\left(net_2^{(3)}(k)\right) \\ O_3^{(3)}(k) = k_D(k) = h\left(net_3^{(3)}(k)\right) \end{cases} \tag{4}$$

$w_{ij}^{(3)}$ is the coefficient of weight from the middle layer to the output layer and the function $h(x) = \frac{e^x}{e^x + e^{-x}}$, which is the nonnegative sigmoid fuction, because the parameters of PID control must be positive.

The bias connecting the real output and the output in expectation of the system is $e(k) = O_i - x_j$, then The PID control value can be calculated using the incremental PID control algorithm. $u(k)$, and the discretization result is shown in equation:

$$\begin{cases} u(k) = k_P\left\{e(k) + \frac{T}{T_I}\sum_{i=1}^{k} e(i) + \frac{T_D}{T}[e(k) - e(k-1)]\right\} \\ u(k-1) = k_P\left\{e(k-1) + \frac{T}{T_I}\sum_{i=1}^{k} e(i) + \frac{T_D}{T}[e(k-1) - e(k-2)]\right\} \end{cases} \tag{5}$$

$T_I$and T_D are the integration and differentiation time parameters, respectively. Further, the control increment of PID can be obtained by equation (2).
The loss function of BP-PID is defined as mse (mean square error)

$$E(k) = \frac{1}{2}\left(O_i - x_j\right)^2 \tag{6}$$

Then, the weights and parameters of the model are updated using the gradient descent method through back propagation:

$$\begin{cases} \Delta w_{li}^{(3)}(k) = \alpha \Delta w_{li}^{(3)}(k-1) + \eta \delta_l^{(3)} O_i^{(2)}(k) \\ \delta_l^{(3)} = e(k) sgn\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right) \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} h^{'\left(net_l^{(3)}(k)\right)} \\ \Delta w_{li}^{(2)}(k) = \alpha \Delta w_{li}^{(2)}(k-1) + \eta \delta_l^{(2)} O_j^{(1)}(k) \\ \delta_l^{(2)} = f^{'\left(net_i^{(2)}(k)\right) \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k)} \end{cases} \tag{7}$$

In the system of equations, $h(x)' = h(x)\left(1 - h(x)\right)$, $f'(x) = (1 - f^2(x))/2$, α is the inertia coefficient, η is the learning rate.

The BP neural network utilizes its self-learning and self-adaptive capabilities to regulate the network's weights, and the parameters of controller can self-adjust to output a better set of parameters for the system. It can improve many problems existing in the face of complex objects using traditional tuning methods, and can significantly enhance the system's control performance.

4. Optimization method based on particle swarm optimization

4.1. Standard particle swarm optimization

The results of the traditional BP neural network heavily depend on the initial value settings of the learning rate η and inertia coefficient α during the parameter iteration and adaptive learning process, which utilizes the gradient descent method. Improper initial value Settings are easy to cause the algorithm to fall into local extrema, which will lead to network oscillation and control performance degradation [5]. In the BP-PID control method, the PSO algorithm was used to globally optimize η and α in the iterative process to ensure that the algorithm converged to the global optimal solution and enhance the overall performance of the algorithm.

PSO method uses a large group of particles, each with a position and velocity, to iteratively adjust their positions based on personal and global best solutions. By exploring the search space in a cooperative manner, particles aim to converge towards the optimal solution.

$$\begin{cases} v_{id}^{k+1} = s v_{id}^k + c_1 r_1\left(p_{id}^k - x_{id}^k\right) + c_2 r_2\left(p_{gd}^k - x_{id}^k\right) \\ x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \end{cases} \tag{8}$$

The values of r_1 and r_2 are random positive numbers within 1. c_1 and c_2 are acceleration constants, which are generally the same values. s is inertia factor. Then, v_{id}^k and x_{id}^k represent the velocity and position of particle i in d-dimension during the KTH iteration. p_{id}^k is the individual extremum of particle i in d-dimension; p_{gd}^k is the global extremum of the PSO in d-dimension.

The rate of convergence of the PSO method is determined by the inertia factor s. The step size of each particle update in the algorithm is directly proportional to the size of s. Consequently, as s increases, the ability to search globally of the method becomes stronger, but the local search ability weakens. Conversely, as s decreases, the ability to search locally becomes stronger, but the global search ability weakens. However, the traditional PSO algorithm has a fixed inertia factor s, which hinders its ability to consider both global and local search capabilities. Therefore, it is necessary to update s during the PSO iteration.

DOI: 10.54254/2753-8818/31/20241088

4.2. Improved PSO algorithm

4.2.1. Update s based on the number of iterations

$$s_t = s_{max}\left(1-\left(\frac{t}{T}\right)\right) \tag{9}$$

t is the number of current iterations, T represents the total iteration number, s_t is the inertia factor corresponding to the TTH iteration step, and smax is the maximum inertia factor of the initial setting [9]. From equation (9), it can be seen that at the beginning of the iteration, the size of t is little, and s_t is big. With the increase of iterations, t is gradually approaching the total iteration number t, and st is decreasing. So that the modified PSO method has both fast rate of convergence and high convergence precision.

4.2.2. Update s based on random weight

$$\omega = \mu + \delta * N(0,1) \tag{10}$$

$$\mu = \mu_{min} + (\mu_{max} - \mu_{min}) * rand(0,1) \tag{11}$$

δ is the value used to represent the deviation degree between the inertia weight ω and its mathematical expectation. The value of δ can control the weight error in the value and make the inertia weight evolve in favor of the expected weight. Additionally, the maximum value μ_{max} is set to 0.8 and the minimum value μ_{min} is set to 0.3 [10].

This method proposes to refine the inertia weight of the algorithm from linear decreasing to random weight. This method can partially avoid the shortcomings of linearly decreasing inertia weight and reduce the probability of falling into local optimum. PSO with random inertia weight strategy is less likely to fall into local extremum.

4.3. Process of PSO-BP network optimizing PID parameter

The optimization of PID parameters using the refined PSO-BP neural network involves the following main steps. Firstly, the initial weights are optimized using the particle swarm optimization algorithm. This algorithm recovers the weight parameter matrix of the neural network from the optimal particle vector. Subsequently, the weights are continuously optimized and adjusted online through the BP neural network. This process continues until the weights reach their optimal values or meet the predefined maximum time limit. Once the weights are optimized, the three parameters of the PID controller are adjusted, completing the overall optimization process. The process of the PID controller, which is based on the PSO-BP network, is illustrated in Figure 4.

DOI: 10.54254/2753-8818/31/20241088

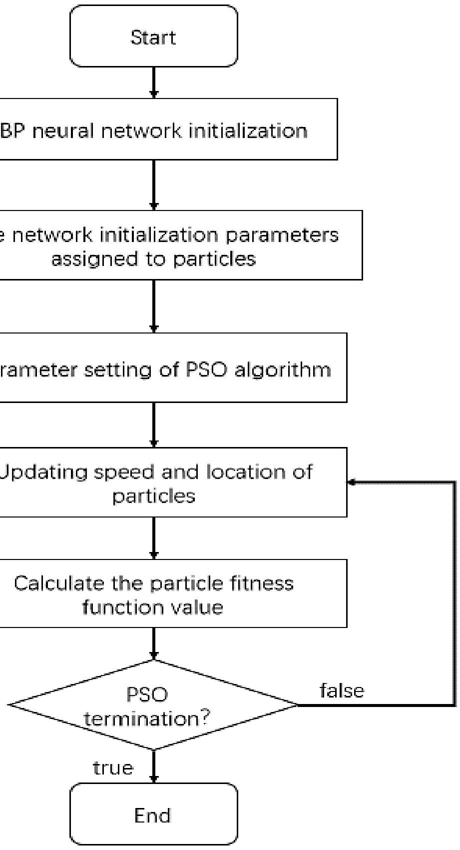

Figure 4. Flow chart of the PSO-BP-PID control algorithm (Picture credit: Original)

Step 1: BP network structure initialization. The learning rate of BP network, initial parameters and inertia coefficient are set.

Step 2: The PSO algorithm initialization involves utilizing the learning rate and inertia coefficient of the system. Additionally, the initial particle population number, particle velocity, and position vector are established.

Step 3: Particle update. According to equation (9), the information of each particle is updated, and the P_i of each particle and the P_g of the whole cluster are calculated.

Step 4: PSO iteration termination judgment. Determine whether the current state meets the iteration termination condition, if so, the iteration terminates, otherwise go back to step 3.

Step 5: When the PSO iteration terminates, the optimal learning rate and inertia coefficient are output as the initial parameters of the BP-PID control system.

5. Discussion

The reasons for choosing BP neutral network and PSO algorithm and their advantages have been introduced above. So, this part the disadvantages will be discussed.

Long training time: The training duration of the BP neural network is considerably extended due to the necessity of a substantial volume of sample data and iterative calculations. This is particularly evident when dealing with intricate control problems.

Sufficient training data is required: BP neural network needs a large number of sample data for training, and for some systems it is difficult to obtain enough sample data, which may affect the performance of the controller.

DOI: 10.54254/2753-8818/31/20241088

Sensitivity to initial solution: The performance of the PSO algorithm is susceptible to the initial solution, and different initial solutions may lead to different search results, so the initial solution needs to be selected carefully.

Large amount of computation: The PSO algorithm needs to carry out a large number of iterative calculations, especially in high-dimensional parameter space and complex systems, its computation may be large.

In summary, the main problem is that the self-learning process of BP neural network needs to consume more computing resources, it puts forward higher requirements for the algorithm running hardware, and the higher computational complexity leads to the general real-time performance of the algorithm. The central objective of the forthcoming investigation revolves around devising strategies to diminish the intricacy of the algorithm while concurrently enhancing its real-time efficiency.

6. Conclusion

The conventional PID controller exhibits certain limitations, including intricate parameter tuning and the inability to be adjusted in real-time online, rendering it unsuitable for controlling time-varying non-stationary systems. To address this issue, the proposed approach integrates the BP neural network with the PID controller, leveraging the adaptive learning capability of the former to dynamically adjust the parameters of the latter. This technique enhances the system's adaptability to time-varying non-stationary systems while simultaneously controlling the system's initial value through BP-PID.

Then, an improved PSO algorithm is proposed to optimize it and ensure its convergence to the global optimal solution. The simulation experiments have demonstrated that the PSO-BP-PID method, when compared to the conventional approach, exhibits superior control accuracy and stability. Moreover, it also exhibits promising potential for practical applications. To conclude, the PID control method based on BP neural network optimized by PSO has strong global optimization ability and parameter adaptive adjustment ability, and can reduce the dependence on sample data. However, this method has some challenges in parameter tuning, initial solution selection and computational burden, which need to be handled carefully. In practical implementation, the selection of a suitable control approach is imperative, taking into account the specific problem's requirements, limitations, and the evaluation of its pros and cons. With the continuous development of artificial intelligence and automation technology, PID control method based on PSO and BP neural network may be more widely used. By combining technologies such as big data analytics, machine learning, and deep learning, automatic modeling, intelligent parameter optimization, and real-time decision-making of control systems can be achieved, thereby improving the performance and adaptability of the system.

In general, the PID control method based on PSO and BP neural network may be further improved and applied in the future. With the continuous development of related technologies and the deepening of application research, this method is expected to play a greater role in the field of automatic and intelligent control, and provide a more effective and stable solution for practical control problems.

References

[1] Gu Z Y, Liu M J and Li J D. Fall detection algorithm based on autoregressive model and neural network [J]. Computer Engineering and Design, 2018, 39(2):537-541.

[2] Jimenez T. An auto-tuning PID control system based on genetic algorithms to provide delay guarantees in passive optical networks [J]. Expert systems with applications, 2015, 42(23):9211-9220.

[3] Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. Heliyon. 2022 May 11;8(5):e09399.

[4] Duan Y M. Research of PID Control Based on BP Neural Network and PSO Algorithm [J]. Computer Technology and development, 2014, 24(8):238-241.

[5] Pirasteh-Moghadam M, Saryazdi MG, Loghman E, E AK, Bakhtiari-Nejad F. Development of neural fractional order PID controller with emulator. ISA Trans. 2020 Nov; 106:293-302.

DOI: 10.54254/2753-8818/31/20241088

[6] Zeng X F. The PID control algorithm based on particle swarm optimization optimized BP neural network [J]. Electronic Design Engineering, 2022, 30(11):69-73+78.
[7] Zhang S F, Song X M and Zhu B H. Brushless DC Motor Controller Combined with Improved PSO-BP Neural Network [J]. Acta Scientiarum Naturalium Universitatis Nankaiensis, 2021, 54(4): 62-67.
[8] Cheng S F, Cheng X H and Yang L. Research onModel and Control System of Brushless DC Motor Based on PSO-BP Neural Network [J]. MICROMOTORS, 2014, 47(08):44-47.
[9] Guo K, San Z and Zhu Y. PID controller parameters optimization based on PSO-BP neural networks [J]. Electronic Design Engineering, 2012, 20(4):63-66.
[10] Zhu X Y and Ma P. PID parameter optimization method based on improved PSO-BP neural network [J]. Modern Electronics Technique, 2022, 45(21):127-130.