Received: 3 September 2021 | Revised: 28 September 2021 | Accepted: 18 October 2021

IET Renewable Power Generation

DOI: 10.1049/rpg2.12320

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Design and analysis of genetic algorithm and BP neural network based PID control for boost converter applied in renewable power generations

Qingsong Wang[1,2] | Haoyu Xi[3] | Fujin Deng[1] | Ming Cheng[1] | Giuseppe Buja[4]

[1] School of Electrical Engineering, Southeast University, Nanjing, China

[2] Jiangsu Key Laboratory of Smart Grid Technology and Equipment, Southeast University, Nanjing, China

[3] School of Software Engineering, Southeast University, Suzhou, China

[4] Department of Industrial Engineering, University of Padova, Padova, Italy

Correspondence
Qingsong Wang, School of Electrical Engineering, Southeast University, Nanjing, China.
Email: qswang@seu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 52177171, 51877040

Abstract

Recently, solar power generation systems are more and more popular and widely used in grid connected power generation, intelligent buildings, and power supply in remote areas. For photovoltaic panels, due to the influence of factors such as light intensity and ambient temperature, their output voltage and current become uns, and the output voltage of a single photovoltaic panel is considerably low. As a result, Boost circuits are needed for voltage boosting. PID controller is commonly used for Boost converter because it can effectively control the controlled object according to the characteristics of the controlled object. However, when the controlled object is complex and variable, the appropriate parameters are hardly to be selected by experience, and the fixed controller parameters may lead to unexpected performances under different working conditions. Here, a genetic algorithm combined with BP neural network PID control (GA-BPPID) is proposed to improve both dynamic and anti-interference performances of Boost circuit by introducing the global optimization ability of genetic algorithm and the adaptive adjustment characteristics of BP neural network. System modelling and detailed controller design procedures are provided. Finally, the theoretical analysis and controller design are validated by simulation results.

1 | INTRODUCTION

Boost converter is commonly used for distributed photovoltaic power generation systems to increase the low module voltage to high load voltage [1]. A typical solar grid connected power generation system is shown in Figure 1. The output power of the photovoltaic panel is connected to the voltage source inverter after passing through the Boost circuit, and the output can be connected to the power grid or directly supply power to the load. Among them, the main functions of Boost circuit include voltage boosting, voltage stability, electrical isolation and current ripple suppression. The purpose of voltage boosting is to change the output voltage of photovoltaic cell into a stable value after power conversion, which is conducive to the operation of power inverter [2]. Electrical isolation is to isolate the photovoltaic panel from the inverter to improve the safety level and reduce interference. Reducing current pulsation can improve the operating performance and lifetime of photovoltaic panels.

Proportional Integral Differential (PID) controller is often used to control the on and off of Boost circuit power switch to achieve the effect of output voltage stabilization because of its simple algorithm and easy implementation [3]. However, when the controlled object is complex and variable, the appropriate parameters of PID controller are difficult to be selected only by experience, and the fixed controller parameters cannot make the controlled object achieve satisfactory performances under different operating conditions.

With the rapid development of artificial intelligence (AI), relevant intelligent technologies are gradually mature. Reinforcement learning is widely used in control and optimization problems, especially in the control of distributed generation [4–7]. Some publications combine AI with traditional PID

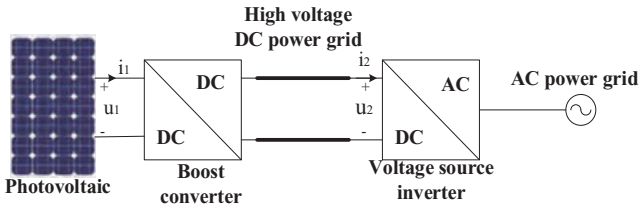

FIGURE 1 Typical photovoltaic power generation system

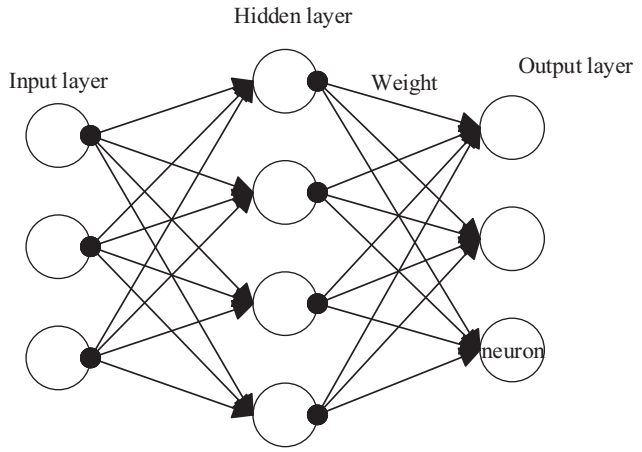

FIGURE 2 Neural network structure

technology to improve the control performances of PID and enrich the diversity of PID control. In [8–10], in order to obtain the correct values of three parameters of PID controller, fuzzy control is combined with traditional PID, where the fuzzy algorithm is used to process the calculated error change rate.

Neural network technology is also an important part of AI technology. Using the characteristics of self-learning and strong adaptability of BP neural network, several publications combine BP neural network with PID technology (BPPID) [11, 12]. BP neural network can self-study and adjust the weight to change the output PID parameters, ensuring strong adaptability of the controller. Radial Basis Function (RBF) neural network has been selected by many literatures to combine with PID controller [13, 14] because of its faster convergence speed. A novel Generalised Hopfield neural network (GHNN) based self-adaptive PID controller for load frequency control (LFC) is also designed in [15].

Intelligent optimization algorithm can solve most global optimization problems, and it is also widely used in PID controller parameter optimization. A genetic algorithm (GA) optimization method of PID controller is proposed in [16] to realize on-line multivariable and multi-objective optimization. Paper [17] presents the random reinforcement genetic algorithm (RR-GA) algorithm to avoid the local optimum efficiently. In [18], a salp swarm algorithm (SSA) is utilised to adjust the fractional-order PID controller coefficients. The results indicate that the proposed controller has fewer frequency variations. In [19], hybrid salp swarm algorithm–simulated annealing (HSSA-SA) algorithm is introduced to enhance the proficiency of PID controller by sensibly plucking the gain parameters, and the supremacy of this method is substantiated.

As analysed above, it is feasible and effective to combine intelligent algorithm with neural network. As one of the main optimization algorithms, particle swarm optimization algorithm is used to optimize the initial weight of neural network and improve the performance of PID controller in controlling complex systems [20, 21].

As discussed, Boost converter is widely used in the photovoltaic power generation system. In this paper, a novel control consists of BP neural network PID control combined with GA is proposed for Boost converter to improve its dynamic response and stability performances. The purpose of GA is to optimize the initial state of BP neural network. By comparing the control effects of traditional PID, BPPID and GA-BPPID, it is revealed that the GA-BPPID controller optimized by GA has the best performance.

Specifically, the organizational structure of this paper is as follows. In Section 2, optimization principle of BP neural network and GA is described. In Section 3, the modelling of Boost circuit is briefly explained. Besides, detailed design process of GA optimization and BP neural network in PID control for Boost converter is well illustrated. In Section 4, the proposed control method is validated by simulation results. In section 5, conclusions are drawn on the paper results.

2 | PRINCIPLE OF BP NEURAL NETWORK AND GA

2.1 | Principle of BP algorithm

The basic principle of BP algorithm is that the learning process is composed of signal forward propagation and error back propagation [22]. The common neural network structure is shown in Figure 2. In forward propagation, the input samples are transferred from the input layer, and then processed by each hidden layer, finally, they will be transmitted to the output layer. If the actual output of the output layer does not conform to the expected output, the error will be backpropagated. Error back propagation is to transmit the output error to the input layer in some form through the hidden layer and allocate the error to all the units of each layer, so as to obtain the error signal of each layer, which is the basis for correcting the unit weight. The weight adjustment process of each layer of signal forward propagation and error back propagation is repeated. The process of weight adjustment is the process of network learning and training. This process continues until the error of network output is reduced to an acceptable level, or until the pre-set learning times.

2.2 | Formula derivation of BP neural network

As shown in Figure 3, the model diagram of a three-layer perceptron is given, the input vector is:

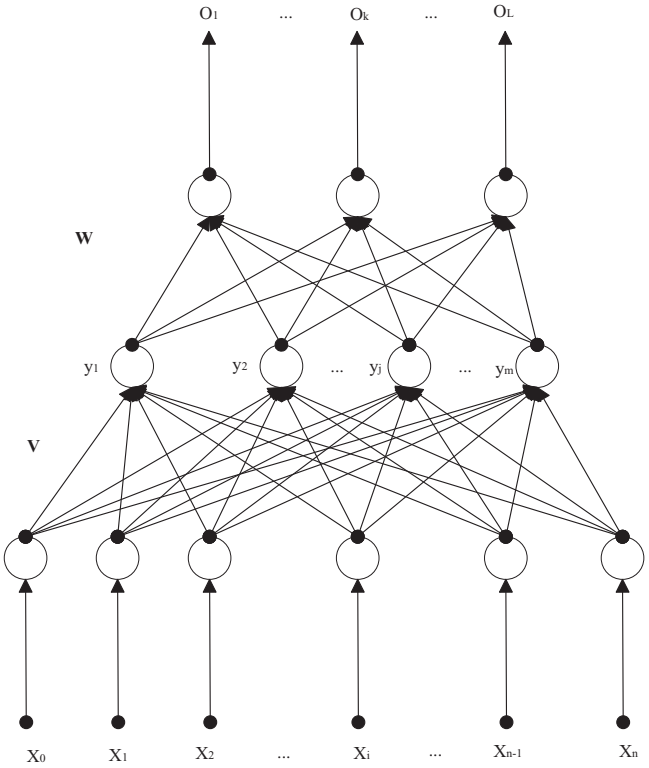

FIGURE 3 Schematic diagram of BP neural network

$$\vec{X} = (x_1, x_2, \ldots, x_i, \ldots x_n)^T \tag{1}$$

The output vector of hidden layer is:

$$\vec{Y} = (y_1, y_2, \ldots, y_j, \ldots y_m)^T \tag{2}$$

The output vector of output layer is:

$$\vec{O} = (o_1, o_2, \ldots, o_k, \ldots o_L)^T \tag{3}$$

The expected vector of output layer is:

$$\vec{d} = (d_1, d_2, \ldots, d_k, \ldots d_L)^T \tag{4}$$

The weight between the input layer and the hidden layer is represented by $\vec{V}$:

$$\vec{V} = \begin{Bmatrix} v_{11}, v_{12}, \ldots, v_{1j}, \ldots v_{1m} \\ v_{21}, v_{22}, \ldots, v_{2j}, \ldots v_{2m} \\ \ldots, \ldots, \ldots, \ldots, \ldots, \ldots \\ v_{i1}, v_{i2}, \ldots, v_{ij}, \ldots, v_{im} \\ \ldots, \ldots, \ldots, \ldots, \ldots, \ldots \\ v_{n1}, v_{n2}, \ldots, v_{nj}, \ldots v_{nm} \end{Bmatrix} \tag{5}$$

The weight between the hidden layer and the output layer is represented by $\vec{W}$:

$$\vec{W} = \begin{Bmatrix} w_{11}, w_{12}, \ldots, w_{1k}, \ldots w_{1L} \\ w_{21}, w_{22}, \ldots, w_{2k}, \ldots w_{2L} \\ \ldots, \ldots, \ldots, \ldots, \ldots, \ldots \\ w_{j1}, w_{j2}, \ldots, w_{jk}, \ldots, w_{jL} \\ \ldots, \ldots, \ldots, \ldots, \ldots, \ldots \\ w_{m1}, w_{m2}, \ldots, w_{mk}, \ldots w_{mL} \end{Bmatrix} \tag{6}$$

And x_0 is the threshold of the network.

Let the activation function of the hidden layer be $f(x)$, and the activation function of the output layer be $g(x)$, the output of the output layer can be expressed as:

$$net_k = \sum_{j=0}^{m} w_{jk} y_j \tag{7}$$

$$o_k = g(net_k) \tag{8}$$

The output expression of the hidden layer is:

$$net_j = \sum_{i=0}^{n} v_{ij} x_i \tag{9}$$

$$y_j = f(net_j) \tag{10}$$

The difference between the actual output and the expected output is error E which can be expressed as:

$$E = \frac{1}{2}(d - O)^2 = \frac{1}{2}\sum_{k=1}^{L}(d_k - O_k)^2 \tag{11}$$

According to Equations (1) and (2), the error E can be expanded to the hidden layer:

$$\begin{aligned} E &= \frac{1}{2}\sum_{k=1}^{L}(d_k - g(net_k))^2 \\ &= \frac{1}{2}\sum_{k=1}^{L}\left[d_k - g\left(\sum_{j=0}^{m} w_{jk} y_j\right)\right]^2 \end{aligned} \tag{12}$$

The error E can also be expanded to input layer:

$$\begin{aligned} E &= \frac{1}{2}\sum_{k=1}^{L}\left\{d_k - g\left[\sum_{j=0}^{m} w_{jk} f(net_j)\right]\right\}^2 \\ &= \frac{1}{2}\sum_{k=1}^{L}\left\{d_k - g\left[\sum_{j=0}^{m} w_{jk} f\left(\sum_{i=0}^{n} v_{ij} x_i\right)\right]\right\}^2 \end{aligned} \tag{13}$$

Next, the gradient descent method is used to update the weights. Since the gradient of a certain point is its first-order partial derivative, the partial derivatives of the error E for the weights w and v are needed to be calculated, respectively.

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} \tag{14}$$

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}} = -\eta \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial v_{ij}} \tag{15}$$

Among them, the sign represents the gradient descent, and η is the learning rate. Choosing the appropriate value of the learning rate is helpful to improve the performance of the neural network.

To make the result more convenient, the error signal errcan be defined in the output layer and the hidden layer respectively, which can be deduced by chain rules

$$err_k^o = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} = -\frac{\partial E}{\partial o_k} g'(net_k) \tag{16}$$

$$err_j^y = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial net_j} = -\frac{\partial E}{\partial y_j} f'(net_j) \tag{17}$$

According to the above formula, the adjustment of the weight vector of the output layer and the hidden layer can be written as:

$$\Delta w_{jk} = \eta \cdot err_k^o \cdot y_j \tag{18}$$

$$\Delta v_{ij} = \eta \cdot err_j^y \cdot x_i \tag{19}$$

According to (1), (2) and (5)

$$\frac{\partial E}{\partial o_k} = -(d_k - o_k) \tag{20}$$

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial y_j} = -\sum_{k=1}^{L} (d_k - o_k) g'(net_k) w_{jk} \tag{21}$$

The results of Equations (16) and (17) can be calculated as:

$$err_k^o = (d_k - o_k) g'(net_k) \tag{22}$$

$$\begin{aligned} err_j^y &= \left[\sum_{k=1}^{L} (d_k - o_k) g'(net_k) w_{jk} \right] f'(net_j) \\ &= \left(\sum_{k=1}^{L} err_k^o \cdot w_{jk} \right) f'(net_j) \end{aligned} \tag{23}$$

Substitute Equations (22) and (23) into Equations (18) and (19) yields:

$$\Delta w_{jk} = \eta \cdot (d_k - o_k) g'(net_k) \cdot y_j \tag{24}$$

$$\Delta v_{ij} = \eta \cdot \left(\sum_{k=1}^{L} err_k^o \cdot w_{jk} \right) f'(net_j) \cdot x_i \tag{25}$$

Using the calculated Equations (22) and (23), the weights of the neural network can be modified after each state update to ensure the optimal output of the neural network.

2.3 | Introduction to the principle of GA

GA is a kind of random search algorithm based on the natural selection and natural genetic mechanism of biology [23], which is a global optimization probability algorithm. It introduces the biological evolution principle of "survival of the fittest and survival of the fittest" into the coding string population formed by the parameters to be optimized and screens each individual according to a certain fitness function and a series of genetic operations. Individuals with high fitness will be retained and form a new group. The new group contains a lot of information from the previous generation and introduces new individuals better than the previous generation. Because of its unique working principle, GA can carry out global optimization search in complex space and has strong robustness. In addition, GA basically does not need any restrictive assumptions for search space, such as continuity, differentiability and so on. Its main features include:

1. The coding strategy of GA directly operates on structural objects, which has a wide range of applications.
2. There are not too many mathematical requirements for the optimization problem to be solved, and there is no need to understand the internal properties of the problem.
3. GA uses random technology to search a coded parameter space efficiently, which is suitable for the search of multimodal distribution. It is not easy to fall into local optimization and easy to parallelize.
4. It is suitable for dealing with complex and nonlinear optimization problems that are difficult to be solved by traditional search methods.

The fitness function in GA often corresponds to the objective function of the optimization problem, the population corresponds to a set of solutions of the optimization problem, and the chromosome corresponds to a feasible solution. Its basic idea is to start from multiple solutions and iterate step by step through certain rules to produce new solutions.

2.4 | GA optimization process

The main process of Optimizing BP neural network by GA can be shown in Figure 4, and the steps are as follows:

1. At the beginning, the topology of the neural network should be determined and the length of the weight to be optimized should be obtained.

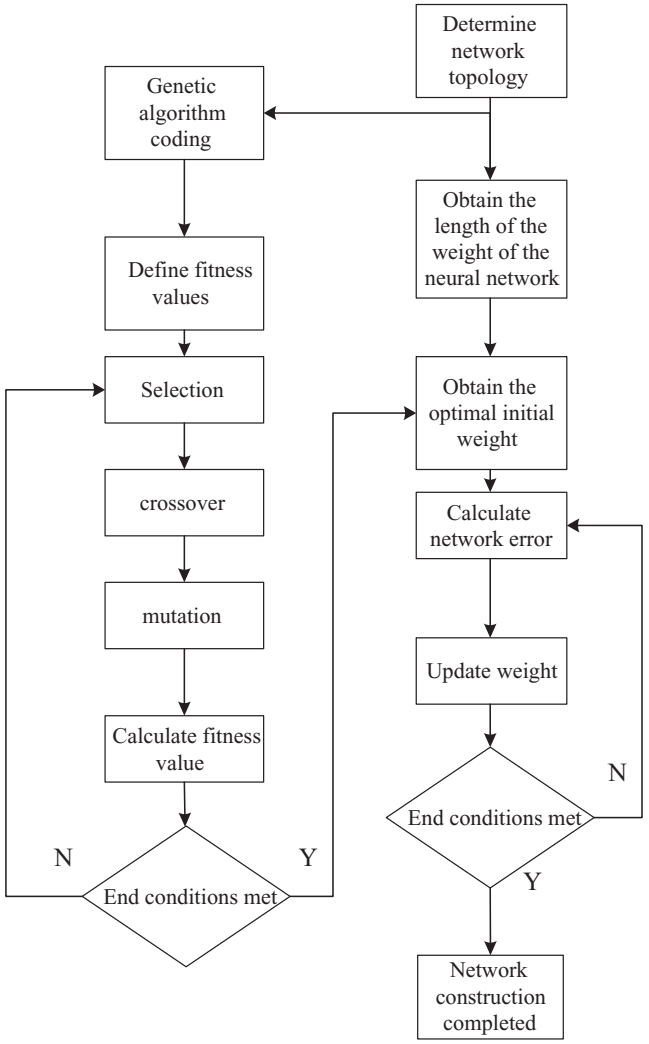

FIGURE 4 Flow chart of BP neural network optimized by GA

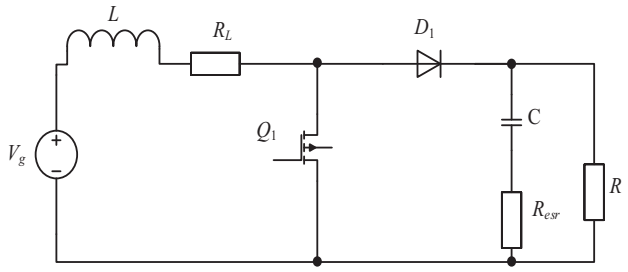

FIGURE 5 Basic diagram of Boost converter

2. The weights of neural network are coded in some form, so that all weights will be represented by genes, and the combination of a string of genes is a solution. Based on coding, these gene strings can be operated by GA.
3. The initialization population includes the range and probability of population crossover, the probability of mutation and the initialization of population individuals. After population initialization, the search algorithm starts to execute.
4. The fitness value of each individual is calculated to judge the viability of the individual.
5. Excellent individuals are selected from the exchanged population, so that they can reproduce the next generation as parents. The principle of choice is that individuals with strong adaptability are more likely to be selected, that is, the principle of survival of the fittest.
6. According to the results of crossover probability calculation, every two selected parents exchange different genes to produce new individuals, which not only improve the diversity of the population, but also reflect the idea of population information exchange.
7. By simulating gene mutation caused by various accidental factors in biology, some genes of individuals in the population are randomly mutated. The new individuals produced by mutation have gene values that the previous generation individuals do not have, which increases the diversity of the population and embodies the idea of individual evolution. In this way, after a new individual is generated, the parents and children are integrated into a new population, and the new population is used to enter the next iteration process.
8. After the above process is iterated for a certain number of times, the final population is output and decoded to obtain the optimized initial weight of BP neural network. The initial weight obtained by this method often has good results.
9. Take the optimized weight as the initial weight of the neural network and update the network with the back-propagation algorithm described above until the requirements are met.

3 | GA-BPPID DESIGN FOR BOOST CONVERTER

3.1 | Modelling of boost converter

Figure 5 shows the diagram of the traditional Boost converter. The high frequency network averaging method is used for modelling. This method is based on averaging the variables of switching elements and using controlled source instead of nonlinear switching elements to form equivalent circuit, and then the transfer function of Boost converter is obtained by small signal method [24] as follows.

$$G_{vd}(s) = \frac{V_g((1-D)^2R - sL)}{(1-D)^2((1-D)^2R + sL + s^2RLC)} \quad (26)$$

3.2 | BP neural network design

The BP neural network combined with PID control belongs to on-line tuning. According to the relevant theory in the second section, in Figure 6 the neural network structure is set as follows: four input layer neurons, five hidden layer neurons and three output layer neurons, in which four input units are reference value $r(k)$, actual value $y(k)$, error value $e(k)$ and threshold

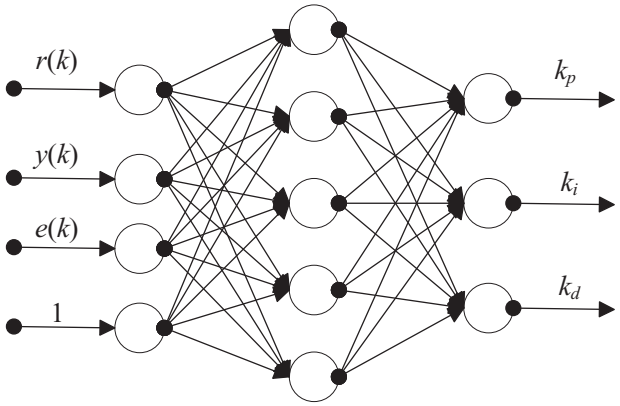

FIGURE 6 Structure of BP neural network

1 respectively, and three output units are three parameters k_p, k_i and k_d of PID respectively.

The activation function of the hidden layer is positive and negative symmetric sigmoid function $f(x)$ which is the source of the nonlinearity of the neural network, and the activation function of the output layer is relu activation function $g(x)$ which can avoid the output value of sigmoid activation function limited to the range of 0 to 1. Besides, it can also accelerate the convergence speed and improve the computational efficiency.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{27}$$

$$g(x) = \max(0, x) \tag{28}$$

3.3 | GA design

After establishing the network topology, genetic algorithm is used to optimize the network performance. The specific operation steps are as follows:

1. The number of weights of BP neural network should be calculated before optimization. There are 4 × 5 weights from the input layer to the hidden layer and 5 × 3 weights from the hidden layer to the output layer, so it is determined that a total of 35 weights need to be optimized.
2. In order to improve the search speed of genetic algorithm and avoid too long individual gene length, the above weights are coded by decimal coding.
3. The fitness function of this genetic algorithm is defined as the difference between the reference value and the real value of PID control signal. The smaller the fitness of an individual, the smaller the error, and the higher the probability of survival of the individual.
4. The selection process uses the selection strategy based on fitness ratio. There are N individuals in the random initial population, each individual is composed of chromosomes, and chromosomes at different positions correspond to different weights. First, their fitness values are calculated for the initial individuals, and then the individuals are randomly selected by roulette algorithm.
5. The crossover process adopts the single point crossover mode, and the crossover probability is defined as P_c, whether to perform crossover operation is determined by randomly generating a number between 0 and 1 and comparing it with P_c. The gene crossover is carried out at the random gene between two individuals to improve the diversity of the population.
6. In the process of mutation, the mutation probability is defined as P_m. Similarly, P_m also determines whether the individual will mutate. Once the conditions are met and each code is mutated according to Equation (29), where X_{new} represents the individual after mutation, X represents the individual before mutation, b is the maximum value boundary of the individual, a is the random number between 0 and 1, n is the current number of iterations, and m is the maximum number of iterations:

$$X_{new} = X + b - X \cdot \left[a \cdot \left(1 - \frac{n}{m}\right)\right]^{2} \tag{29}$$

7. The operation of the above GA is iterated m times, and the optimal initial weight is obtained after iterations. Then the value is used to start the training of BP neural network. By using the gradient descent method to update the weight, the on-line tuning of PID controller parameters can be realized.

TABLE 1 Parameters for simulations

Symbol	Quantity	Values
L	inductance	0.08 H
R	resistance	5 Ω
C	capacitance	0.007 F
V_g	supply voltage	5 V
V_{ref}	reference voltage	7 V
D	duty cycle	0.285714
L_r	learning rate	0.001
α	inertia factor	0.05
N	number of individuals	10
P_c	crossover probability	0.2
P_m	mutation probability	0.1
m	maximum number of iterations	120

4 | SIMULATIONS AND DISCUSSIONS

4.1 | Parameter setting

To test the performance of GA-BPPID, simulation verification is carried out on MATLAB/Simulink. The parameters for simulations are shown in Table 1.

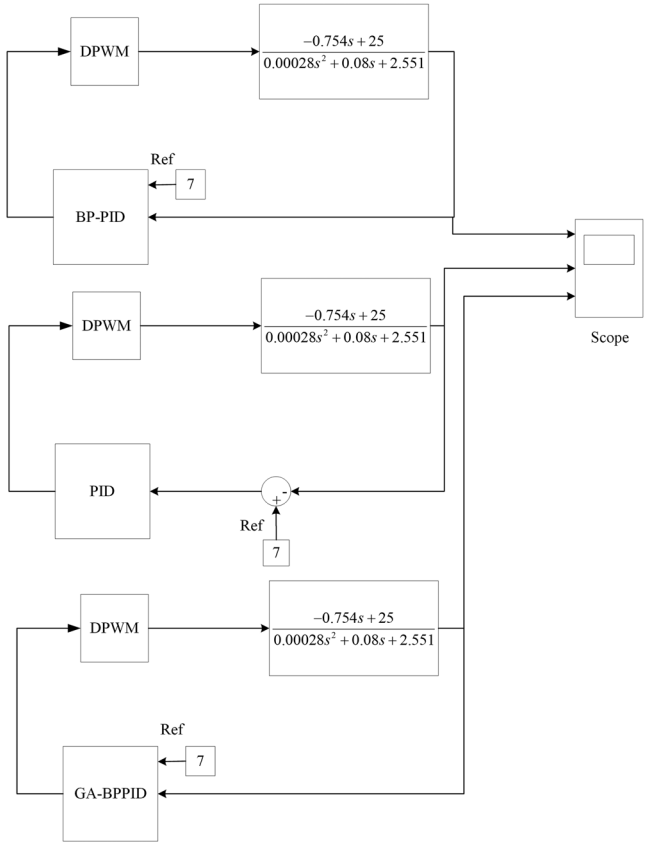

FIGURE 7 Main circuit of the proposed controller

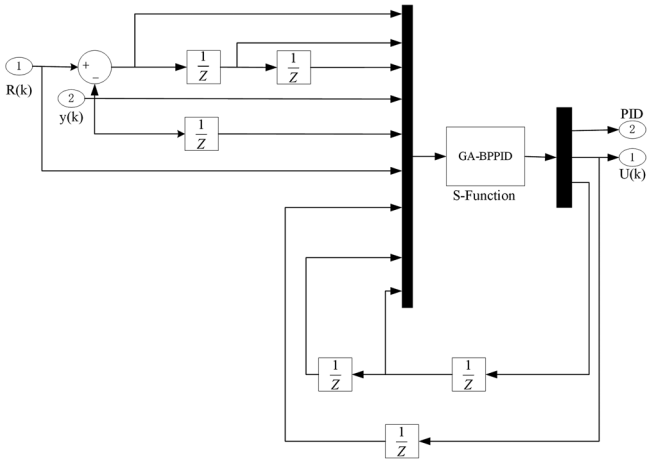

FIGURE 8 GA-BPPID circuit based on S-function module

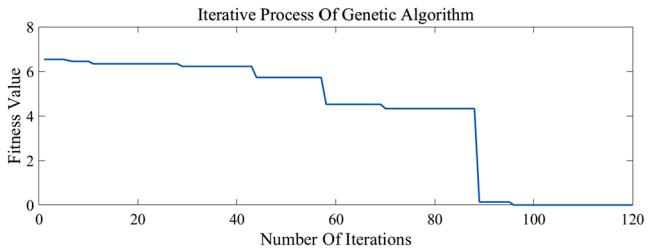

FIGURE 9 Iterative process of GA

4.2 | Simulations and results

Through the parameters of the Boost circuit set above, the transfer function of the Boost circuit can be calculated, which is taken as the controlled object to simulate under PID control, BPPID control and GA-BPPID control respectively. In Figure 7, the above three control methods are compared based on the same transfer function obtained from the Boost circuit, where BPPID control is on the upper side, PID control in the middle and GA-BPPID on the lower side.

Figure 8 is the GA-BPPID circuit diagram written by S-function module. Figure 9 is the optimization iteration process of GA. The fitness value of *y*-axis reflects the error between the actual value and the reference value of the output of the PID controller, while *x*-axis represents the number of iterations of GA. It can be seen from the figure that when the GA is iterated nearly 90 times, the fitness value at this time is very small, indicating that the weight at this time is in a better state of optimization.

The output voltage of the Boost circuit under the three control methods is observed in Figure 10. All the control circuits can stably output 7 V reference voltage after a certain period. Obviously, the control effect of BP neural network combined with PID controller is better than that of traditional PID controller, and the performance of BPPID controller optimized by GA is significantly improved compared with the other two.

The dynamic performance indexes of these three waveforms in Figure 10 are shown in the Table 2.

By comparing the data in the table, it can be clearly seen that the overshoot values of the Boost circuit voltage waveform output by BPPID controller and GA-BPPID controller are far less than 29.66% under traditional PID control. In addition, BPPID successfully increased the adjustment time from 0.406 s of traditional PID to 0.343 s, and GA-BPPID again shortened the adjustment time to 0.136 s by optimizing neural network.

Figure 11 shows the changes of the weights of the internal neural networks of BPPID and GA-BPPID. Figure 11a shows the weight change process of BPPID controller. There are 35 curves in the figure, and each curve represents the change process of 35 weights in the neural network. Figure 11b shows the weight change process of GA-BPPID controller. Similarly, the

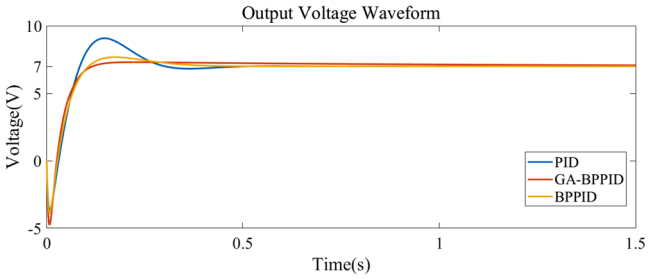

FIGURE 10 Comparison of output voltage waveform under PID control, BPPID control and GA-BPPID control

TABLE 2 Dynamic performance indexes

Symbol	Quantity	PID	BPPID	GA-BPPID
t_d	delay time	0.052s	0.051s	0.046s
t_r	rise time	0.084s	0.107s	0.195s
t_p	peak time	0.147s	0.174s	
t_s	settle time	0.406s	0.343s	0.136s
$\sigma\%$	overshoot	29.66%	9.68%	0%

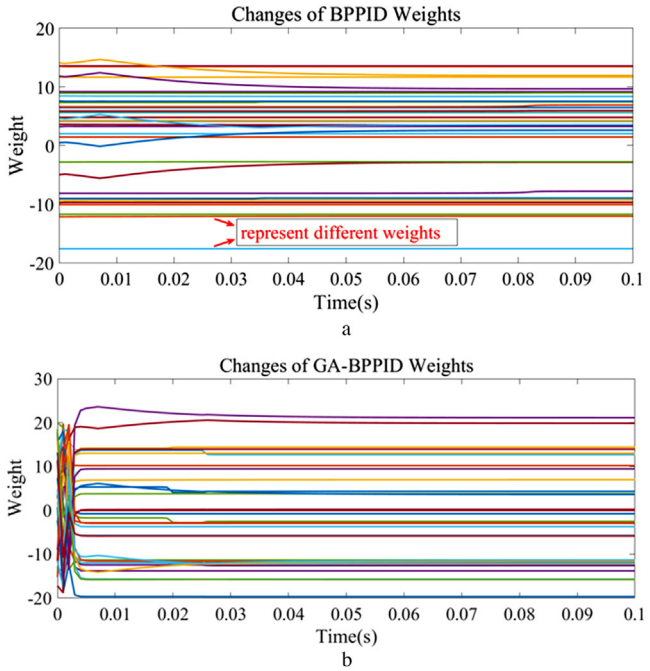

FIGURE 11 Weight changes of different controllers

figure also reflects the change process of 35 weights. The difference is that GA is used for global optimization search in the initial state, so the initial weights are still in the search state with a large change range ratio, and all weights are set between −20 and 20 in the process of GA optimization.

The anti-interference performances of traditional PID, BPPID and GA-BPPID controllers are tested. At the simulation time of 0.9 s, interferences with amplitude of 2 and duration of 0.01 s are added on the three controller circuits, and the output voltage waveforms of the three circuits after interference are observed. As shown in the results of the three waveforms in Figure 12, the traditional PID can recover to the target state in the face of interference by its fixed parameters, but it takes a long time, and the waveform fluctuation is relatively large. BPPID can use neural network to realize on-line adjustment of PID parameters, and the corresponding waveform is relatively stable, and the fluctuation range is small. GA-BPPID optimizes the weight based on BPPID, which makes the controller adjust parameters more quickly and accurately, and it can greatly improve the anti-interference performance, which is the best among the three methods.

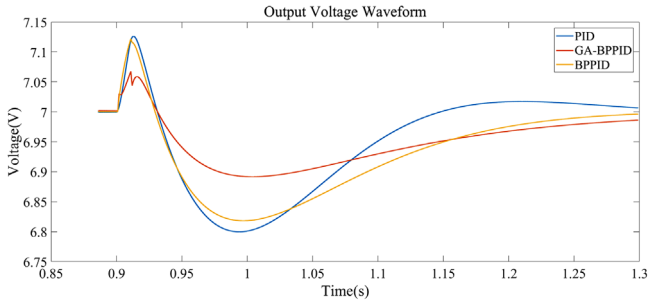

FIGURE 12 Comparison of anti-interference performance under PID control, BPPID control and GA-BPPID control

5 | CONCLUSION

This paper presents an adaptive controller based on GA to optimize BP neural network and combined with traditional PID control. The system modelling and detailed design process are provided. The three-layer BP neural network selected in the design can adjust the weight of each layer according to the gradient descent method. Compared with the traditional PID controller, which can only rely on experience to set fixed controller parameters, BPPID and GA-BPPID can adjust the controller parameters more flexibly. Especially when dealing with complex objects, GA-BPPID further optimizes the weight based on BPPID, making the controller more capable of online adjustment. By observing the output voltage of Boost circuit under three different controllers and the waveform results under certain interferences, it is found that the proposed GA-BPPID controller not only greatly reduce the overshoot and settle time of output voltage, but also improve the stability and anti-interference performance of the system.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under project 52177171 and 51877040.

CONFLICT OF INTEREST

No.

DATA AVAILABILITY STATEMENT

Author elects to not share data.

ORCID

Qingsong Wang https://orcid.org/0000-0002-0066-9973
Haoyu Xi https://orcid.org/0000-0001-8922-6716
Fujin Deng https://orcid.org/0000-0002-9832-004X
Ming Cheng https://orcid.org/0000-0002-3466-234X
Giuseppe Buja https://orcid.org/0000-0003-4245-1742

REFERENCES

1. Bellinaso, L.V., Figueira, H.H., Basquera, M.F., et al.: Cascade control with adaptive voltage controller applied to photovoltaic Boost converters. IEEE Trans. Ind. App. 55(2), 1903–1912 (2019)
2. Lee, K., Park, B., Kim, R., et al.: Robust predictive current controller based on a disturbance estimator in a three-phase grid-connected inverter. IEEE Trans. Power Electron. 27(1), 276–283 (2012)

3. Chen, L., Chen, G., Wu, R., et al.: Variable coefficient fractional-order PID controller and its application to a SEPIC device. IET Control Theory App. 14(6), 900–908 (2020)
4. Cao, D., Hu, W., Zhao, J., et al.: Reinforcement learning and its applications in modern power and energy systems: a review. J. Mod. Power Syst. Clean Energy 8(6), 1029–1042 (2020)
5. Cao, D., Hu, W., Zhao, J., et al.: A multi-agent deep reinforcement learning based voltage regulation using coordinated PV inverters. IEEE Trans. Power Syst. 35(5), 4120–4123 (2020)
6. Wang, W., Gao, X., Fan, B., et al.: Faulty phase detection method under single-line-to-ground fault considering distributed parameters asymmetry and line impedance in distribution networks. IEEE Trans. Power Delivery 1–1 (2021). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463787 https://doi.org/10.1109/TPWRD.2021.3091646
7. Li, S., Hu, W., Cao, D., et al.: Electric vehicle charging management based on deep reinforcement learning. J. Mod. Power Syst. Clean Energy 1–12 (2021). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9465776 https://doi.org/10.35833/MPCE.2020.000460
8. Nayak, N., Mishra, S., Sharma, D., et al.: Application of modified sine cosine algorithm to optimally design PID/fuzzy-PID controllers to deal with AGC issues in deregulated power system. IET Gener. Transm. Distrib. 13(12), 2474–2487 (2019)
9. Abazari, A., Monsef, H., Wu, B.: Load frequency control by de-loaded wind farm using the optimal fuzzy-based PID droop controller. IET Renew. Power Gener. 13(1), 180–190 (2019)
10. Shen, J., Xin, B., Cui, H., et al.: Control of single-axis rotation INS by tracking differentiator based fuzzy PID. IEEE Trans. Aerosp. Electron. Syst. 53(6), 2976–2986 (2017)
11. Alizadeh, M., Kojori, S.S., Ganjefar, S.: A modular neural block to enhance power system stability. IEEE Trans. Power Syst. 28(4), 4849–4856 (2013)
12. Wang, L., Zhan, Z., Xin, Y., et al.: Development of BP neural network PID controller and its application on autonomous emergency braking system. In: Proceedings of the IEEE Intelligent Vehicles Symposium, Changshu, China, pp. 1711–1716 (2018)
13. Nie, L., Guan, J., Lu, C., et al.: Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network. IET Intell. Transp Syst. 12(6), 485–494 (2018)
14. Kumar, M.B.H., Saravanan, B., Sanjeevikumar, P., et al.: Review on control techniques and methodologies for maximum power extraction from wind energy systems. IET Renew. Power Gener. 12(14), 1609–1622 (2018)
15. Ramachandran, R., Madasamy, B., Veerasamy, V., et al.: Load frequency control of a dynamic interconnected power system using generalised Hopfield neural network based self-adaptive PID controller. IET Gener. Transm. Distrib. 12(21), 5713–5722 (2018)
16. Ren, H., Fan, J., Kaynak, O.: Optimal design of a fractional-order proportional-integer-differential controller for a pneumatic position servo system. IEEE Trans. Ind. Electron. 66(8), 6220–6229 (2019)
17. Cao, Y., Wang, Z., Liu, F., et al.: Bio-inspired speed curve optimization and sliding mode tracking control for subway trains. IEEE Trans. Veh. Technol. 68(7), 6331–6342 (2019)
18. Babaei, F., Lashkari, Z.B., Safari, A., et al.: Salp swarm algorithm-based fractional-order PID controller for LFC systems in the presence of delayed EV aggregators. IET Electr. Syst. Trans. 10(3), 259–267 (2020)
19. Nayak, J.R., Shaw, B., Sahu, B.K.: Implementation of hybrid SSA–SA based three-degree-of-freedom fractional-order PID controller for AGC of a two-area power system integrated with small hydro plants. IET Gener. Transm. Distrib. 14(13), 2430–2440 (2020)
20. Liu, H., Zhao, Y., Shurafa, M.A., et al.: A novel PID control strategy based on PSO-BP neural network for phase-shifted full-bridge current-doubler synchronous rectifying converter. In: Proceedings of the IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, Chongqing, China, pp. 1241–1245 (2021)
21. Yuan, J., Wang, R., Jiang, L.: Research on neural network PID adaptive control with industrial welding robot in multi-degree of freedom. In: Proceedings of IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, pp. 280–284 (2016)
22. Esfandiari, K., Abdollahi, F., Talebi, H.A.: Adaptive control of uncertain nonaffine nonlinear systems with input saturation using neural networks. IEEE Trans. Neur. Net Lear. 26(10), 2311–2322 (2015)
23. Sun, Y., Bing, X., Zhang, M., et al.: Automatically designing CNN architectures using the genetic algorithm for image classification. IEEE Trans. Cyber. 50(9), 3840–3854 (2020)
24. Ayachit, A., Kazimierczuk, M.K.: Averaged small-signal model of PWM DC-DC converters in CCM including switching power loss. IEEE Trans. Circuits Syst. II. 66(2), 262-266 (2019)

How to cite this article: Wang, Q., Xi, H., Deng, F., Cheng, M., Buja, G.: Design and analysis of genetic algorithm and BP neural network based PID control for boost converter applied in renewable power generations. IET Renew. Power Gener. 16, 1336–1344 (2022). https://doi.org/10.1049/rpg2.12320