Pitch Control of Wind Turbines Based on BP Neural Network PI

Jing Du*, Bo Wang

School of Mechanical Engineering, Chongqing University, Chongqing, 400044, China

*Corresponding author’s e-mail: dujing@cqu.edu.cn

Abstract. For wind turbine operating above the rated wind speed, the output power of the generator is maintained near the rated power through pitch control. Due to the non-linear relationship between the pitch angle and the wind speed, the traditional PI controller is not ideal for control above the rated wind speed. Therefore, the currently used PI controller generally incorporates gain scheduling technology. A wind turbine pitch controller based on BP neural network PI is proposed in this paper, which optimizes PI parameters and has better control effects. First, the PI control strategy with gain scheduling is briefly introduced, and then the principle and implementation steps of BP neural network PI are given. Finally, compared the control effects of the two control strategies of PI with gain scheduling and BP neural network PI, a new co-simulation between Simpack and MATLAB/Simulink is built, and it is proved that the BP neural network PI can improve the effect of pitch control.

1. Introduction

Wind energy is gradually replacing traditional fossil energy due to its clean, pollution-free and abundance. With the improvement of the degree of automation of wind turbines, people have begun to pursue high wind energy utilization coefficient and output power curve while improving the reliability of wind turbine operation [1]. When the wind speed is less than the rated wind speed, the pitch angle is kept unchanged, and the wind speed is adjusted by speed control to obtain the optimal wind energy utilization coefficient, thereby obtaining more wind energy. when the wind speed is greater than the rated speed, the pitch angle is adjusted by pitch control to make output power near the rated power.

With the maturity of wind power technology, high-power wind turbines have occupied a dominant position in the wind power market. The use of high-power wind turbines will inevitably lead to further increase in the size of the impeller and nacelle, and the operating environment of the wind turbine is extremely harsh, and the wind speed has the characteristics of instability and randomness. Therefore, the traditional PI controller cannot stabilize the output well. power. Based on this, scholars at home and abroad have conducted a lot of optimization research on PI pitch controller. Zheng [2] proposed a pitch control method based on neuron PID, which stabilized the generator output power and reduced the fatigue load of wind turbine, but did not consider the dynamic changes of the pitch control system. Jafamejadsani et al. [3] proposed an adaptive control method based on radial basis function (RBF) neural network, which improved the control effect, but the blade pitch was larger. Tian et al. [4] proposed a sliding mode variable structure pitch strategy based on RBF neural network, which improved the power performance of the system and reduced the pitch angle jitter. Zhang et al. [5] proposed a GA-PID control scheme to optimize PID parameters and improve the robustness of the system.
The PI pitch control strategy based on BP neural network proposed here can effectively improve the control effect of the system. First, the PI control strategy with gain scheduling is briefly introduced, and then the basic implementation steps of BP neural network PI is given in detail. Finally, compared the control effects of the two control strategies of PI with gain scheduling and BP neural network PI, a new co-simulation between Simpack and MATLAB/Simulink is built to verify the effectiveness of the given control strategy.

2. PI control strategy with gain scheduling

At present, PI pitch controllers are commonly used, and due to the time-varying and instability of wind speed, the generator speed is generally used as the measured value. Specifically, the rated speed of the generator is used as the input signal, and the filtered generator speed is used as the feedback signal. Compared the two signals, the deviation value obtained is multiplied by the gain of the PI controller to calculate the pitch angle. Since the relationship of pitch angle and wind speed are non-linear, the pitch angle that needs to be changed to maintain stable output power at different wind speeds is very different, and the constant PI gain is not enough to achieve effective speed control. Therefore, when the operation of wind turbine is at different wind speeds, different gains need to be used to adjust the proportional coefficient and integral coefficient in the original PI controller, that is, gain scheduling [6, 7]. The structure diagram of PI controller with gain scheduling is as follows.

![Figure 1. PI controller with gain scheduling.](image1)

3. BP neural network PI control strategy

The BP neural network is a network formed by multiple neurons connected to each other in a specific way. It has a strong nonlinear mapping ability. Through the self-learning of the network system, the PI controller parameters are optimized to achieve the desired control effect. Combining BP neural network with PI controller, the control block diagram is as follows. In the figure, \( x_{in} \) is the rated speed of the generator, \( y_{out} \) is the measured generator speed, error is the difference between the measured generator speed and the rated speed, \( k_p \) and \( k_i \) are the two parameters of the PI controller, and \( u \) is the output of the controller. By inputting the rated speed, the measured speed and the speed difference into the BP neural network, the weight of the neuron is adjusted online, so as to realize the adaptive adjustment of the PI controller parameters [8].

![Figure 2. BP neural network PI controller.](image2)
The BP neural network in this article is a 3-layer feedforward network with hidden layers, where the input layer is 3, the hidden layer is 5, and the output layer is 2. The input of BP neural network is

\[ O^i_0(n) = x_i(n) \quad i = 1, 2, 3 \]  

(1)

Where, \( i \) is the number of input variables. The upper right corner of the variable is marked with 0,1,2, which represent the input layer, hidden layer and output layer respectively. The input variables of the PI controller are

\[ x_i(n) = e(n) - e(n-1) \]  

(2)

\[ x_2(n) = e(n) \]  

(3)

Where, \( e(n) \) is the difference between the expected output and the actual output at the current sampling moment, that is

\[ e(n) = x_m(n) - y_{oa}(n) \]  

(4)

The input and output of the hidden layer are

\[ v^j_i(n) = \sum_{o=0}^{3} w^i_j(n)O^o_i(n) \]  

(5)

\[ O^j_1(n) = \varphi(v^j_i(n)) \quad j = 1, 2, 3, 4, 5 \]  

(6)

Where, \( w^i_j \) is the weight of the hidden layer, \( \varphi(x) \) is the activation function of the hidden layer, and the hyperbolic tangent function is often used. That is

\[ \varphi(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]  

(7)

The input and output of the output layer are

\[ v^j_2(n) = \sum_{o=0}^{5} w^2_j(n)O^o_j(n) \]  

(8)

\[ O^k_2(n) = f(v^k_i(n)) \quad k = 1, 2 \]  

(9)

Where, \( O^k_1(n) = k_p \), \( O^k_2(n) = k_i \). \( w^2_j \) is the weight of the output layer, and \( f(x) \) is the activation function of the output layer.

Since the two outputs of the output layer correspond to the two parameters \( k_p \) and \( k_i \) of the PI controller, and neither of them can be negative, the activation function of the output layer takes a non-negative sigmoid function, that is

\[ f(x) = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}} \]  

(10)

From equation (8), the control law of the incremental PI controller is
\[ u(n) = u(n-1) + \Delta u(n) \]
\[ = u(n-1) + k_1 [e(n) - e(n-1)] + k_2 e(n) \]
\[ = u(n-1) + \sum_{k=1}^{2} x_k(n) O_k^2(n) \]  
(11)

Where, \( u(n) \) is the control quantity at the current sampling moment.

Taking the quadratic power of the output error as the performance index, the performance index function is

\[ \varepsilon(n) = \frac{1}{2} e^2(n) \]  
(12)

When using the gradient descent method to modify the weight of the network, in order to accelerate the convergence speed of the learning process and reduce the numerical oscillation, a momentum term that makes the search quickly converge to the global minimum will be introduced.

\[ \Delta w_k^2(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_k^2(n)} + \alpha \Delta w_k^2(n-1) \]
\[ = \eta \delta_k^2(n) O_k^2(n) + \alpha \Delta w_k^2(n-1) \]  
(13)

Where \( \eta \) is the learning rate and \( \alpha \) is the momentum factor.

Make

\[ \delta_k^2(n) = -\frac{\partial \varepsilon(n)}{\partial w_k^2(n)} \]
\[ = -\frac{\partial \varepsilon(n)}{\partial y_{out}(n)} \frac{\partial y_{out}(n)}{\partial \Delta u(n)} \frac{\partial \Delta u(n)}{\partial O_k^2(n)} \frac{\partial O_k^2(n)}{\partial v_k^2(n)} \]  
(14)

From equation (4), equation (9) and equation (11), we can get

\[ \frac{\partial \varepsilon(n)}{\partial y_{out}(n)} = -(y_{in}(n) - y_{out}(n)) = -e(n) \]  
(15)

\[ \frac{\partial O_k^2(n)}{\partial v_k^2(n)} = f'(v_k^2(n)) \]  
(16)

\[ \frac{\partial \Delta u(n)}{\partial O_k^2(n)} = x_k(n) \]  
(17)

\[ \frac{\partial y_{out}(n)}{\partial \Delta u(n)} = \text{sgn} \left( \frac{\partial y_{out}(n)}{\partial \Delta u(n)} \right) \]

Since \( \frac{\partial y_{out}(n)}{\partial \Delta u(n)} \) is unknown, the symbolic function \( \text{sgn} \left( \frac{\partial y_{out}(n)}{\partial \Delta u(n)} \right) \) is generally used instead, and the learning rate is adjusted to compensate for the inaccuracy of the calculation.

In summary

\[ \delta_k^2(n) = e(n) \text{sgn} \left( \frac{\partial y_{out}(n)}{\partial \Delta u(n)} \right) x_k(n) f'(v_k^2(n)) \]  
(18)

The calculation formula for the weight correction of the network output layer is
\[ \Delta w_{ij}(n) = \eta \delta_j(n)O_j(n) + \alpha \Delta w_{ij}(n-1) \]  \hfill (19)

The calculation formula of network hidden layer weight correction is

\[ \Delta w_{ji}(n) = \eta \delta_i(n)O_i(n) + \alpha \Delta w_{ji}(n-1) \]  \hfill (20)

among them

\[ \delta_i(n) = \varphi' \left( v_i(n) \right) \sum_{k=1}^{2} \delta_k(n)w_{ki}(n) \]  \hfill (21)

4. Simulation analysis

Simpack is a software for studying multi-body dynamics simulation analysis. It is widely used in the wind turbine industry with its efficient modelling method, fast and stable solver, etc. [9]. This paper uses MATLAB/Simulink and Simpack co-simulation to compare and analyze PI with gain scheduling and BP neural network PI. The model used in the simulation is the 5MW offshore wind turbine model provided by NREL [10]. The specific parameters are as follows.

| Parameter                  | Parameter value |
|----------------------------|-----------------|
| Rated power                | 5 MW            |
| Hub height                 | 90 m            |
| Hub diameter               | 126 m           |
| Rated wind speed           | 11.4 m/s        |
| Rated speed of Hub         | 12.1 r/min      |
| Generator rated speed      | 1173.7 r/min    |

Since this article mainly studies the pitch control strategy of the wind turbine operating above the rated wind speed, turbulent wind is used to simulate wind conditions, with an average wind speed of 20m/s and a simulation time of 60s. As shown in Figure 4.

![Figure 3. Turbulent wind.](image)

The physical model of the wind turbine can be built in Simpack, and the control strategy of the wind turbine pitch system can be defined in Simulink, and data is exchanged in real time via the co-simulation interface SIMAT between Simpack and MATLAB/Simulink. The following figure is the joint simulation model of the wind turbine pitch control system block diagram.
Figure 4. Simulink model of joint simulation of wind turbine pitch control system.
The Simulink model with gain scheduling PI and the Simulink model based on BP neural network PI are shown in the figure below.

![Simulink model with gain scheduling PI](image1)

**Figure 5.** Simulink model with gain scheduling PI.

![Simulink model of BP neural network PI](image2)

**Figure 6.** Simulink model of BP neural network PI.

Figure 7, 8, 9 and 10 are the curves of pitch angle, output power, torque and generator speed of the control strategy of PI with gain scheduling and PI control based on BP neural network respectively. It can be seen from the figure that the two control strategies can maintain the stability of the output power above the rated wind speed, but the BP neural network PI pitch control strategy has a better control effect.
Table 2. Output power data.

| Control strategy                  | Max   | Min   | Std   |
|----------------------------------|-------|-------|-------|
| PI with gain scheduling          | 5005  | 4995  | 2.207 |
| BP neural network PI             | 5004  | 4994  | 1.92  |

Table 3. Torque data.

| Control strategy                  | Max     | Min     | Std     |
|----------------------------------|---------|---------|---------|
| PI with gain scheduling          | 4.45e4  | 4.142e4 | 870.1   |
| BP neural network PI             | 4.424e4 | 4.157e4 | 698.6   |

Table 4. Generator rated speed data.

| Control strategy                  | Max   | Min   | Std   |
|----------------------------------|-------|-------|-------|
| PI with gain scheduling          | 1225  | 1132  | 24.53 |
| BP neural network PI             | 1221  | 1138  | 19.78 |
Table 2, 3, and 4 are the output power, torque and generator speed data respectively. It can be seen from the data that compared with PI pitch control with gain scheduling, the output power of the two is not much different, but the standard deviation of the PI output power of the BP neural network has dropped by 13.0%, and the standard deviation of the generator torque has dropped. The standard deviation of the generator speed has also dropped by 19.4%, and the maximum speed has not exceeded the rated speed. Therefore, the PI pitch control strategy based on the BP neural network must have a better control effect.

5. Conclusions
For the wind speeds above the rated wind speed, the pitch control is needed to use to prevent excessive output power from damaging the wind turbine. Based on this, a PI pitch control strategy based on BP neural network is proposed, and then the co-simulation between Simpack and MATLAB/Simulink is built. Compared the PI with gain scheduling with the BP neural network PI pitch control strategy, the results show that the BP neural network PI pitch control strategy can effectively maintain the generator output power and speed stability. It needs to be noted that the model used in the simulation in this paper is a simplified model, and when modeling multi-body dynamics, only the degree of freedom of each component in one direction is considered. In addition, the flexibility of important components should also be taken into consideration.

Acknowledgments
The paper is sponsored by Natural Science Foundation of China (approval No.:51975066). The Authors acknowledge assistance or encouragement from colleagues and valuable comments from reviewers.

Reference
[1] Ren, L.N, Li, M.Y, Wang, Z.C. (2015) Application of Improved Bee Colony Algorithm in Variable Pitch Control of Wind Turbine. Machinery Design and Manufacture, 53(3): 44.
[2] Zheng, Y. (2010) Independent Pitch Control of Wind Turbine Generator Based on Neuron PID. Water Resources and Power, 30(1):151-154.
[3] Jafarnejadsani, H, Pieper, J, Ehlers, J. (2013) Adaptive Control of a Variable-Speed Variable-Pitch Wind Turbine Using Radial-Basis Function Neural Network. IEEE Transactions on Control Systems Technology, 21(6):2264-2272.
[4] Tian, M, Zhang, B.W, Zhou, L.W, et.al. (2019) Research on Independent Pitch Control Based on RBF Neural Network Sliding Mode Variable Structure. Power system protection and control, 047(004):107-114.
[5] Zhang, Z.W, Wang, P.Y, An, B.N, et.al. (2017) Pitch Control of Wind Turbines Based on Genetic Algorithm PID. Power Electronics, 51(07):37-39+85.
[6] Xie, S.Y, Jin, X, Chen, J. (2014) Parameter design of wind turbine pitch controller based on FAST. Wind turbine technology, (6):74-77.
[7] Lescher, F, Zhao, L.Y, Borne, P. (2005) Robust Gain Scheduling Controller for Pitch Regulated Variable Speed Wind Turbine. Studies in Informatics and control, 14(4):299-315.
[8] Liu, J.K. (2004) Advanced PID Control MATLAB Simulation. Electronic industry press, Beijing.
[9] Zhou, S.X. (2013) SIMPACK 9 example tutorial. Beijing united publishing company, Beijing.
[10] Jonkman, J, Butterfield, S, Musial, W, et.al. (2009) Definition of a 5 MW Reference Wind Turbine for Offshore System Development. Springfield: Technical Report NREL/ TP-500-38060, National Renewable Energy Laboratory.