LQR optimized BP neural network PI controller for speed control of brushless DC motor

Tingting Wang¹, Hongzi Wang¹,², Huangshui Hu³ and Chuhang Wang⁴

Abstract
This paper proposes a linear quadratic regulator (LQR) optimized back propagation neural network (BPNN) PI controller called LN-PI for the speed control of brushless direct current (BLDC) motor. The controller adopts BPNN to adjust the gain $K_p$ and $K_i$ of PI, which improves the dynamic characteristics and robustness of the controller. Moreover, LQR is adopted to optimize the output of BPNN so as to make it close to the target PI gains. Finally, the optimized control output is inputted into the BLDC motor system to achieve speed control. The performance analysis of the proposed controller is presented to compare with traditional PI controller, neural network PI controller and LQR optimized PI controller under MATLAB/Simulink, the results shows that the proposed controller effectively improves the response speed, reduces the steady-state error and enhances the anti-interference ability.

Keywords
Linear quadratic regulator, back propagation neural network, PI, BLDC motor, speed control

Date received: 6 July 2020; accepted: 2 October 2020

Handling Editor: James Baldwin

Introduction
Brushless direct current (BLDC) motor has been widely used in aerospace, robotics, electric vehicles and other fields because of its advantages such as simple structure, high efficiency, low maintenance, and high dynamic response.¹⁻³ As is known to all, there are many aspects of motor control, such as sensorless control,⁴ current control,⁵ torque control⁶ and speed control,² etc. In order to achieve accurate speed and position control applications, speed regulation is an important aspect in the driving field of BLDC motor.³⁻⁷ With the continuous rapid development of modern power electronic technology, sensor technology, automatic control technology and manufacturing technology, it is of great practical significance and application prospects to study speed controllers of BLDC motor with fast response, strong regulation ability and high control precision. However, the complex and strong coupling non-linear characteristics make it hard to achieve good performance for speed control by using traditional control methods. Therefore, many new controllers for speed control have been proposed to enhance the performance of BLDC motor in the last few decades.²⁻⁹⁻¹²

¹College of Mechatronic Engineering, Changchun University of Technology, Changchun, China
²College of Computer Science and Engineering, Changchun University of Technology, Changchun, China
³College of Computer Science and Engineering, Jilin University of Architecture and Technology, Changchun, China
⁴College of Computer Science and Technology, Changchun Normal University, Changchun, China

Corresponding author:
Huangshui Hu, College of Computer Science and Engineering, Jilin University of Architecture and Technology, 1111 Xuejian Road, Kuancheng District, Changchun 130114, China.
Email: huhs08@163.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
In order to improve the transient and steady-state characteristics of BLDC motor, proportional integral derivative (PID) controllers are often the best choice for BLDC motor speed regulation. Aiming at the speed control of BLDC motor, a new PI controller variable speed pre-standard value control strategy is proposed. Experiments show that the controller can accurately adjust the speed in a wide range, with low cost and good stability. In Joice et al., a PI-based controller has been developed for speed control of BLDC motor, and the speed controller increased sensitivity and reduced the speed overshooting by increasing the proportional gain. However, the dependence of traditional PID on gain selection seriously affects the performance of BLDC motor, and there are various uncertainties and nonlinearities. Therefore, a variety of PID tuning methods are proposed, such as fuzzy logic, sliding mode control, genetic algorithm, particle swarm optimization, gray wolf optimization algorithm, etc. However, the sliding mode controller inevitably has the chattering problem, which leads to the system performance degradation. The initial population of the genetic algorithm PID controller is not suitable. Particle swarm optimization and gray wolf optimization tend to lead to slow convergence rate and local optimization, while the fuzzy logic PID controller has poor coarseness. The reason for non-adaptation is to extract and adjust the fuzzy rules from the input/output data without providing an online adaptive learning method. In addition, the PID gain tuning process is a time consuming process, so many methods have been proposed to simplify or improve the PID gain tuning process. Neural network based algorithms have reported promising results.

Neural network based PID gain update algorithms have been successfully implemented to control a servo motor, computerized numerical control machine tools, and so on. In Xia et al., a single neuron PI controller has been developed for the control of the BLDC motor. In Premkumar and Manikandan, an adaptive neuro-fuzzy inference system tuned PID speed controller has been presented for BLDC motor. The proposed controller effectively tunes the gain of the PID controller. In Premkumar and Manikandan, Online fuzzy logic supervised learning of radial basis function neural network based speed controller for BLDC motor was presented. This controller receives the supervised error by comparing the Fuzzy PID supervised algorithm output and output of the controller. Based on this supervised error, the parameter of RBFNN is updated in the network. Vikas et al. proposed a neural network-based PID-like controller composed of a mixed locally recurrent neural network and contains at most three hidden nodes which form a PID-like structure. The presented controller is easy to implement. But the main drawback in such strategy is in the number of parameters to be determined prior to the training which requires a priori process knowledge. Moreover, the training algorithm based on gradient descent is a time consuming process. So some methods have been proposed to optimize the process. In Kang et al., the particle swarm optimization algorithm is used to initialize the weight of the adaptive PID neural network controller, and the improved gradient descent algorithm is used to adjust the parameters of the PID neural network. The disadvantage of this method is that the particle swarm optimization algorithm initializes the PID neural network for a longer time. After decades of development, the classical optimal control theory has formed a well-known Linear Quadratic Regulator that minimizes the deviation of the system state trajectory and requires minimal controller effort. This typical behavior of LQR has prompted control designers to use it to adjust PID controllers. Although LQR optimized PID controller has good tracking performance and stable performance, its robustness still needs to be improved.

In view of the difficulty of determining the parameters of the neural network and the poor robustness of LQR, a new LQR optimized BP neural network PI controller which is called LN-PI is designed to achieve the speed control of BLDC motor in this paper. The main contributions of this paper are listed as follows:

1. BPNN is used to constantly adjust the gains of the PI controller according to the motion state of the BLDCM control system, so as to achieve the optimization of performance indexes, make the output state of the output neurons correspond to the gain values of the PI controller, which improves the dynamic characteristics and robustness of the controller.

2. A new LQR optimized BP neural network PI controller is proposed, which takes into account the problems of slow convergence speed and local extremes of BP neural network. LQR is used to optimize the output of BPNN so as to make it faster and closer to the optimal target PI controller gains.

3. The performance of LN-PI controller is evaluated through simulation and compared with conventional PI(C-PI) controller, neural network PI(NN-PI) controller and LQR optimized PI(LQR-PI) controller at different working conditions. The results show that LN-PI controller is superior to those related controllers in aspects such as overshoot, undershoot, stability time, recovery time and steady-state error.

4. The sensitivity of the LN-PI controller is analyzed under the operating conditions of parameter change of the inertia and the permanent magnetic flux of the BLDC motor. The stability and feasibility of the LN-PI controller are verified.
The rest of this paper is organized as follows: section “The speed controller of the BLDC motor” introduces the speed controller of BLDC motor, section “The proposed controller for BLDC motor” designs the LQR optimized BP neural network PI controller in detail, and simulation results are shown in section “Simulation results,” conclusions are outlined in section “Conclusion.”

**The speed controller of the BLDC motor**

Three phase star connected BLDC motor can be transformed into the circuit diagram shown in Figure 1. And the mathematical model of BLDC motor can be described by the following equation in matrix form,

\[
\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} \frac{-R}{L-M} & 0 & 0 & 0 \\ 0 & \frac{-R}{L-M} & 0 & 0 \\ 0 & 0 & \frac{-R}{L-M} & 0 \\ 0 & 0 & 0 & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega_r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{L-M}{L-M} \\ 0 \\ 0 \\ \frac{1}{L-M} \end{bmatrix} + \begin{bmatrix} \frac{1}{L-M} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \\ T_L \end{bmatrix}.
\]

where \( V_a, V_b, V_c \) are the phase voltages in volts, stator winding resistance is presented by \( R \) in ohms, \( i_a, i_b, i_c \) denote the phase currents of the motor in amperes. \( L, M \) in Henry denote the self-inductance of the motor winding and the mutual inductances between stator windings respectively. \( e_a, e_b, e_c \) represent the back electromotive force of each phase in volts. The number of poles in the rotor is represented by \( P \), and \( \theta_r \) is the rotor position of the rotor in radians. \( J, B, \omega_r, T_L \) denote the moment of inertia, frictional coefficient, angular velocity and load torque of the motor respectively. Moreover, the electromechanical torque is given in equation (2):

\[
T_e = B\omega_r + J\frac{d\omega_r}{dt} + T_L
\]

Based on the model of BLDC motor described in Figure 1, the speed control system for BLDC motor is presented in Figure 2.

The system consists of three phase voltage source PWM inverter, three phase BLDC motor, LQR optimized BP neural network, conventional PI controller, switching logic and motor measurement sensors. Furthermore, two closed loops namely inner loop and outer loop are established based on the above blocks. In particular, the outer loop is used for controlling the speed of the BLDC motor, and based on error, the LQR optimized BP neural network PI controller provides the control signal to the switching logic circuit so as to control the dc bus voltage thus by controlling the speed of the motor. The PI controller consists of proportional and derivative terms, which can be expressed as follows:

\[
u(t) = K_P e(t) + K_I \int e(t) dt
\]

where \( e(t), K_P, K_I, u(t) \) are the tracking error, proportional gain, integral gain, and control input, respectively. By taking the time-derivative of both sides of the continuous time PI equation and discretizing the resulting equation, we can obtain the discrete-time PI controller which can be expressed as:

\[
u(k) = u(k-1) + K_P [e(k) - e(k-1)]
+ \frac{K_I}{T_s} [e(k) + e(k-1)]
\]

where, \( T_s \) is the sampling period, \( k \) represents the running step of the discrete PI controller.

**The proposed controller for BLDC motor**

**Design of BP neural network**

This paper adopts a three-layer BP neural network, in which the input layer consists of 3 neurons (The
number of neurons corresponds to $e(k), e(k - e(k - 1), e(k) - 2e(k - 1) + e(k - 2))$, the hidden layer contains six neurons (The selection of the number of neurons is the best for multiple simulations.), and the output layer is made up of two neurons (The number of neurons corresponds to two gains of PI controller.). The structure is shown in Figure 3.

Figure 3. The structure of BP neural network.

In Figure 3, $x_i$ is the actual input of the network, $o_{out}(k = 1, 2)$ is the actual output of the network, $o_{out}(k = 1, 2)$ is the network target output, and $\Delta E, w_{ij}, w_{jk}, \theta_j, \theta_k$ are the control error, the first layer weight, the second layer weight, the bias of the $jth$ hidden layer neurons, the bias of the $kth$ output layer neurons, respectively. Then, the mapping relationship between the input and output of the network is shown as follows:

$$I_j = \sum_{i=1}^{3} w_{ij} \times x_i - \theta_j$$  \hspace{1cm} (5)

$$b_j = f_1(I_j)$$ \hspace{1cm} (6)

$$I_k = \sum_{i=1}^{6} w_{jk} \times b_j(k) - \theta_k$$ \hspace{1cm} (7)

$$o_{out} = f_2(I_k)$$ \hspace{1cm} (8)

Then the optimized PI gain can obtained

$$K_{P, i} = o_{out}(k = 1, 2)$$ \hspace{1cm} (9)

where $f_1(x), f_2(x)$ is the activation function of the hidden layer and output layer, and sigmoid function is shown as follows:

$$f_1(x) = \frac{2}{1 + e^x} - 1$$ \hspace{1cm} (10)

$$f_2(x) = \frac{1}{1 + e^x}$$ \hspace{1cm} (11)

The optimization process of LQR

The design goal of the optimal control system is to minimize the error when the system tracks a certain command signal. The solution to this problem is:

Suppose a discrete system, its discrete equation of state is

$$X(k + 1) = FX(k) + GO_{out}(k)$$ \hspace{1cm} (12)

$$O_k(k) = CX(k)$$ \hspace{1cm} (13)

where $X(k)$ is the $(n \times l)$-dimensional state vector, $O_{out}(k)$ is the $m$-dimensional control vector, $O_k(k)$ is the $l$-dimensional output vector, and $F$, $G$ and $C$ are $(n \times n)$, $(n \times m)$, $(l \times n)$ dimensional system parameter matrix.
Assume that under a stable and appreciable command signal $O_{\delta k}$, the control input $O_{\text{kout}}(k)$ is

$$O_{\text{kout}}(k) = -MX(k) + KE(k)$$

$$E(k + 1) = E(k) + (O_k' - O_{\text{kout}}(k))$$

where $M$ and $K$ are the feedback gain and feedforward gain matrices of $(m \times n)$ and $(m \times l)$ dimensions, which correspond to the error back propagation matrix and the output layer output matrix of BP neural network respectively, and $O_{\delta k}$ is a given instruction vector.

Suppose the system has a steady state equilibrium, which satisfies the stability of equations (12) and (13).

That is,

$$\bar{x} = F\bar{x} + G\bar{o}_{\text{kout}}$$

$$\bar{o}_{\text{kout}} = -M\bar{x} + KE$$

$$\bar{E}_k = E_k(k) - \bar{E}, \bar{o}_k(k) = o_k(k) - \bar{o}_k$$

If a new variable with a “*” is used to indicate that the system deviates from a steady state stationary state value (disturbance), then these new variables are defined as

$$x^*(k) = x(k) - \bar{x}$$

$$o^*_{\text{kout}}(k) = o_{\text{kout}}(k) - \bar{o}_{\text{kout}}$$

$$E^*(k) = E(k) - \bar{E}, o_k^*(k) = o_k(k) - \bar{o}_k$$

Substitute equations (19), (20), and (21) into equations (12) and (13) to obtain:

$$x^*(k + 1) + \bar{x} = F[x^* + \bar{x}] + G[\bar{o}^*_{\text{kout}}(k) + \bar{o}^*_{\text{kout}}]$$

$$o_k^*(k) + \bar{o}_k = c[x^*(k) + \bar{x}]$$

Obviously, by subtracting the steady-state relations (16) and (18) from equations (22) and (23), a new difference equation can be obtained:

$$x^*(k + 1) = Fx^* + Go^*_{\text{kout}}(k)$$

$$o^*_k(k) = cx^*(k)$$

$$o^*_{\text{kout}}(k) = -Mx^*(k) + KE^*(k)$$

And there is

$$o^*_{\text{kout}}(k + 1) = -Mx^*(k + 1) + KE^*(k + 1)$$

$$= -M[Fx^*(k) + Go^*_k(k)] + KE^*(k) + K[\bar{E}_k(k) + (O_k' - O_{\text{kout}}(k))]$$

$$= -MFX^*(k) - MGo^*_{\text{kout}}(k) + KE^*(k) + K[\bar{E}_k - cx(k)]$$

$$= -MFX^*(k) - MGo^*_k(k) + o^*_k(k) + Mx^*(k) - KcE^*(k)$$

$$= (-MF + M - KC)x^*(k) + (I_m - MG)o^*_{\text{kout}}(k)$$

A new augmentation system can be formed by equations (24) and (27),

$$\begin{bmatrix} x^*(k + 1) \\ o^*_{\text{kout}}(k + 1) \end{bmatrix} = \begin{bmatrix} F & G \\ -MF + M - KC & I_m - MG \end{bmatrix} \begin{bmatrix} x^*(k) \\ o^*_{\text{kout}}(k) \end{bmatrix}$$

$$= \begin{bmatrix} F & G \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x^*(k) \\ o^*_{\text{kout}}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} V^*(k)$$

Where

$$V^*(k) = - ([M \hspace{0.2cm} K] [F - I_n \hspace{0.2cm} G] [0 \hspace{0.2cm} I_m]) \begin{bmatrix} x^*(k) \\ o^*_{\text{kout}}(k) \end{bmatrix}$$

Here $V^*(k)$ is an m-dimensional vector. Seeking control input $V^*(k)$ to minimize the following performance index $J$,

$$J = \frac{1}{2} \sum_{k = 0}^{\infty} (x^*(k)^TQx^*(k) + V^*(k)^TRV^*(k))$$

where $Q = CTSC$ is a semi-positive definite matrix of $(n \times n)$ dimensions, and $R$ is a positive definite matrix of $(m \times m)$ dimensions. Under the constraint of equation (28), the performance index represented by equation (30) is minimized. Finding $V^*(k)$ at this point is the same as the linear optimal regulator problem. Therefore, the optimal controller input $V^*(k)$ of the augmentation system can be obtained as

$$V^*(k) = - (R + [0 \hspace{0.2cm} I_m] \hat{P} [0 \hspace{0.2cm} I_m])^{-1} \begin{bmatrix} 0 \\ I_m \end{bmatrix} \begin{bmatrix} F \\ G \end{bmatrix} \begin{bmatrix} x^*(k) \\ o^*_{\text{kout}}(k) \end{bmatrix}$$

where $\hat{P}$ is the Riccati equation of the augmentation system and is a $(n + m) \times (n + m)$ dimensional positive definite matrix.

**Design of LN-PI Controller**

BP neural network has the advantages of strong parallel processing capability, strong nonlinear mapping ability and good fault tolerance. It has been used to adjust the gain of PID controller with different structures. However, BP neural network is difficult to achieve the global optimal solution, the convergence time is short, the parameters are difficult to determine, resulting in low precision and weak convergence. Therefore, introducing LQR into BPNN reduces the control error and obtains the optimal solution.
Let BPNN input be \( x_i(k) \), output \( O_{\text{out}}(k) \), and the control output at this time should be

\[
O_k = O_k^0(k) + O_{\text{out}}(k)
\]

\[
= -Mx_k(k) + KE(k) + O_{\text{out}}(k)
\]

(32)

where \( g(x_i(k)) = g(x_1, x_2, x_3) \) is the input and output nonlinear mapping function of BPNN.

Then, the final optimized PI gain can be expressed as

\[
K_{P,T} = o_k(k = 1, 2)
\]

(33)

Defining the Error Function of Training BPNN Weights, then the dynamic response of the controlled object can be accurately expressed by (12), the output of the BPNN network is zero \( (O_{\text{out}}(k) = 0) \).

This paper defines BPNN’s performance indicators as

\[
J_k = \frac{1}{2}
\]

\[
\left[ x^*(k + 1)^T \quad o^*_{\text{out}}(k + 1)^T \right] P \left[ x^*(k) \quad o^*_{\text{out}}(k) \right] + r^*(k)^T R r^*(k)
\]

(34)

where

\[
x^*(k + 1) = x_i(k + 1) - \pi
\]

(35)

\[
r^*(k) = o^*_{\text{out}}(k + 1)\]

(36)

Connection weights \( W_{ji}(k) \) and \( W_{jk}(k) \) of BPNN are adjusted by the steepest descent method:

\[
W_{ji}(k + 1) = W_{ji}(k) - \eta \frac{\partial J_k}{\partial W_{ji}}
\]

(37)

\[
W_{jk}(k + 1) = W_{jk}(k) - \eta \frac{\partial J_k}{\partial W_{jk}}
\]

Where \( \eta \) is the learning rate.

Calculate \( \frac{\partial J_k}{\partial W_{jk}} \).

\[
= \left( \left[ x^*(k + 1)^T \quad o^*_{\text{out}}(k + 1)^T \right] P \left[ x^*(k) \quad o^*_{\text{out}}(k) \right] + r^*(k)^T R r^*(k) \right)
\]

(38)

\[
= \delta_k \frac{\partial o^*_{\text{out}}(k)}{\partial W_{jk}} = \delta_k f_2(\lambda_k) \cdot b_j(k)
\]

(39)

In the same way, \( \frac{\partial J_k}{\partial W_{ji}} \) can be calculated, and the entire correction formula of the BPNN connection weight is obtained:

\[
W_{ji}(k + 1) = W_{ji}(k) - \eta \delta_k \bullet b_j(k)
\]

(40)

\[
W_{jk}(k + 1) = W_{jk}(k) - \eta \delta_k \bullet x_i(k)
\]

where \( x_i(k) \) represents the input of the BPNN, \( b_j \) is the hidden layer output, and \( O_{\text{out}} \) is the output of the output layer.

The specific steps of the BPNN type PID control algorithm based on LQR optimization are shown in Figure 4:

1. The structure of the BPNN is selected in advance, that is, the number of input layer nodes \( m \) and the number of hidden layer nodes \( q \) are selected, and the initial values \( W_{ji}(0), W_{jk}(0) \),

**Figure 4.** The flowchart of LN-PI control algorithm.
of the weighting coefficients of each layer are randomly generated between $[-1, 1]$, select the learning rate $\eta$ and let $k = 1$.

(2) Sampling to obtain $r(k)$ and $y(k)$, and calculate $e(k) = r(k) - y(k)$.

(3) Normalize $e(k)$, $e(k) - e(k - 1)$, $e(k) - 2e(k - 1) + e(k - 2)$ as an input to BPNN.

(4) Calculate the output $O_{out}$ of the BPNN output layer according to equation (3), and calculate the control output at this time as $O_k$, which is the two adjustable parameters $K_P, K_I$ of the controller.

(5) Calculation of control error $E(k)$ of BPNN based on LQR optimization.

(6) The weighting coefficients $W_{yj}(k), W_{yk}(k)$ of the modified output layer and the hidden layer are calculated by equation (40).

(7) Let $k = k + 1$ and return (2) until the BPNN output layer output error meets the requirements.

(8) Input the final control output into the BLDC motor system to achieve speed control.

Simulation results

The LN-PI controller is designed to further verify its speed control performance in the BLDC motor control system. The Matlab/Simulink toolbox is used to build the system model, and the controller is simulated and compared under different industrial control conditions. The simulation model of the BLDC motor control system is shown in Figure 5(a), the LN-PI controller is shown in Figure 5(b), and the specifications of the BLDC motor used in this paper like those in Ramya et al., Afrasiabi and Yazdi, and Hu et al. are given in Table 1.

The input speed of a given BLDC motor is 2500 rpm, the number of BP neural network nodes is 3–6–2, and the initial weight is $[-0.5, 0.5]$. The weight is dynamically adjusted online with the system, and the learning rate is $\eta = 0.05$, $R = 1$. The experimental tests
obtained the $K_P$ and $K_I$ parameters of the four methods are shown in Table 2:

Firstly, the tests are performed under the condition of no load. The comparison of speed response and current response is shown in Figures 6 and 7 respectively, and the performance indexes are given in Table 3.

As can be seen from Figure 6 and Table 3, the overshoot and undershoot of the C-PI controller are up to 314.46 and 90.86 rpm, while the LQR-PI undershoot is at least 3.07 rpm. The LN-PI controller has a minimum of 10.98 rpm. In the steady state process, the settling time of the C-PI controller is up to 0.081 s and the steady state error is the largest. The NN-PI and LQR-PI controllers are medium in terms of both stability time and steady state error. Horizontally, the LN-PI controller has a minimum convergence time of 0.015 s and a minimum steady state error. In addition, seen from Figure 7, the oscillation phenomena of C-PI and NN-PI are obvious and have a long duration, and LQR-PI and LN-PI have a fast stable speed. Overall, the performance of LN-PI controller is the best, which has the characteristics of fast response, small steady-state error and strong robustness.

Secondly, for the actual BLDC motor system, there will always be sudden external disturbances that are difficult to control. In the research of control algorithms, the robustness of the system is also an important indicator to judge the advantages and disadvantages of the control strategy, and has good anti-interference performance. The control strategy is what is needed in actual engineering control. In order to study the control effect of the anti-interference performance of these four control algorithms, 3 N external force interference is added to the system at 0.2 s. The comparison of speed response and current response is shown in Figures 8 and 9 respectively, and the performance indexes are given in Table 4.

It can be seen from Figure 8 and Table 4 that when the system suddenly increases the external force interference in 0.2 s, the four controllers generate the volatility response, and overshoot/undershoot occurs in the C-PI, NN-PI and LQR-PI controllers. Obviously, the

![Figure 6. Comparison of speed response with no load.](image-url)

![Table 2. $K_P/K_I$ experimental parameters.](image-url)

| Parameters/methods | C-PI | NN-PI | LQR-PI | LN-PI |
|--------------------|------|-------|--------|-------|
| $K_P$              | 0.01 | 0.07  | 0.08   | 2.91  |
| $K_I$              | 30.11| 30.01 | 39.98  | 54.01 |

![Table 3. The performance indexes under the condition of no load.](image-url)

| Control system parameters | Controllers |
|---------------------------|-------------|
|                            | C-PI  | NN-PI | LQR-PI | LN-PI |
| Peak overshoot (rpm)      | 314.46| 120.72| 30.17  | 10.98 |
| Peak undershoot (rpm)     | 90.86 | 20.97 | 3.07   | 33.09 |
| Settling time (s)         | 0.081 | 0.057 | 0.042  | 0.015 |
| Steady state error (rpm)  | 1.03  | 0.87  | 0.47   | 0.32  |
| Steady state error (%)    | 0.041 | 0.035 | 0.019  | 0.013 |
Figure 7. Comparison of current response with no load.

Figure 8. Comparison of speed response under varying load condition.

Table 4. The performance indexes under the condition of varying load condition.

| Control system parameters | Controllers | C-PI | NN-PI | LQR-PI | LN-PI |
|--------------------------|-------------|------|-------|--------|-------|
| Peak overshoot (rpm)     | 35.01       | 16.96| 8.86  | -3.78  |
| Peak undershoot (rpm)    | 136.05      | 113.85| 108.94| 24.81  |
| Recovery time (s)        | 0.071       | 0.066| 0.053 | 0.032  |
| Steady state error (rpm) | 1.22        | 0.79 | 0.65  | 0.41   |
| Steady state error (%)   | 0.049       | 0.032| 0.026 | 0.016  |
fluctuation of LN-PI is the smallest, the recovery time of the LN-PI controller is as short as 0.032 s, and the steady-state error is at least 0.41 rpm. It can be seen from Figure 9 that the oscillations of the C-PI, NN-PI, and LQR-PI controllers are obvious. Although the LN-PI controller has a higher oscillation amplitude, it can quickly restore the stability effect. Therefore, it is proved that the LN-PI controller has the best robustness and good anti-interference ability.

Thirdly, in the actual working process, BLDC often needs to set different speeds to meet the needs of actual work. In order to study the speed control transition process, the simulation study of follow-up control by setting different speeds mainly studies the control effect of the transition process in which the set value changes. The control effects of these four control algorithms are compared mainly from the aspects of transient response speed, stability control accuracy and transient oscillation. The simulated target speed changes from 2500 to 1500 rpm and then to 2500 rpm. The comparison of speed response and current response is shown in Figures 10 and 11 respectively, and the performance indexes are given in Table 5.

It can be seen from Figure 10 and Table 5 that when the speed of 0.2 and 0.4 s changes, the response speed of the LN-PI controller is the fastest, taking 0.2 s as an example. The fluctuation of the C-PI controller is the largest. The overshooting and undershoot are 76.53 and 289.11 rpm, the LQR-PI controller has a minimum overshoot of 4.22 rpm, the LN-PI controller has a minimum undershoot of 18.63 rpm, and the LN-PI controller has the fastest recovery time. The value is 0.031 s and the steady state error is 0.40 rpm. Furthermore, it can also be seen from Figure 11 that when the speed changes, the current response oscillation time of the LN-PI controller is the shortest. Overall, the LN-PI controller has the best tracking performance when speed changes.

Under the above three operating conditions, the LN-PI controller is superior to other controllers. Next, the sensitivity of the LN-PI controller is analyzed under the operating conditions of parameter change of the inertia and the permanent magnetic flux of the BLDC motor. The inertia varies from 100%, 200%, 300%, and 400% of the nominal inertia. The permanent magnetic flux varies from 50%, 100%, 150%, and 200% of the nominal. The sensitivity of the LN-PI controller is shown in Figure 12. It can be seen from the figure that, with the change of inertia, the speed response of BLDC motor is only affected by parameters such as overshoot and undershoot, but will not affect the final system output. When the permanent magnet flux changes, the speed response of BLDC motor is only affected by the steady state error, and the other parameters do not affect the system output. In addition, the variation of parameters such as overshoot and undershoot and steady-state error is not greater than 0.4% of nominal value. From the sensitivity analysis, the LN-PI controller is a suitable controller for wide operating conditions of the BLDC motor.

Conclusion

A LQR optimized BP neural network PI controller is proposed for speed control of brushless dc motor. BP neural network is adopted to adjust the gains of PID
Figure 10. Comparison of speed response under varying set speed condition.

Figure 11. Comparison of current response under varying set speed condition.

Table 5. The performance indexes for varying set speed condition.

| Control system parameters | Controllers |
|---------------------------|-------------|
|                           | C-PI  | NN-PI | LQR-PI | LN-PI |
| Peak overshoot (rpm)      | 76.53 | 28.82 | 4.22   | 52.47 |
| Peak undershoot (rpm)     | 289.11 | 171.76 | 60.85  | 18.63 |
| Recovery time (s)         | 0.63  | 0.056 | 0.044  | 0.031 |
| Steady state error (rpm)  | 1.31  | 1.02  | 0.75   | 0.40  |
| Steady state error (%)    | 0.087 | 0.068 | 0.05   | 0.027 |
controller, and LQR is adopted to optimize the output of BPNN. The whole control system is established and simulated by MATLAB/Simulink, and the effectiveness of the controller is tested under the conditions of no load, with load and variable speed. The performance is compared with C-PI, NN-PI and LQR-PI controllers. Under no load conditions, LN-PI controller is proved to have the minimum overshoot, the shortest settling time and the minimum steady-state error of speed response whose values are 10.98 rpm, 0.015 s, 0.32 rpm, respectively. Compared with C-PI, NN-PI and LQR-PI, the settling time of LN-PI is increased by 81.48%, 73.68%, 64.29%, and the steady-state error is reduced by 68.93%, 63.22%, 31.91%. Under conditions with load, LN-PI controller has the minimum peak undershoot 24.81 rpm, the shortest recovery time 0.032 s, the minimum steady-state error 0.41 rpm. Compared with C-PI, NN-PI and LQR-PI, the recovery time of LN-PI is increased by 54.93%, 51.52%, 39.62%, and the steady-state error is reduced by 66.39%, 48.10%, 36.92%. Under different set speed conditions, LN-PI controller has the shortest recovery time and the minimum steady-state error compared with other controllers under various working conditions of BLDC motor. Moreover, the sensitivity of LN-PI controller is analyzed under the condition of inertia and permanent magnetic flux parameters of BLDC motor, and the stability and feasibility of the control method are verified.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work supported by Capital construction funds in the provincial budget of Jilin development and reform commission in 2019 (grant no 2019C054-4); National nature fund project (grant no 61803044); Science and technology development project of Jilin Province (grant no 201908032005ZG); Science and technology plan project of Jilin Province (20200201009JC).

ORCID iD
Tingting Wang https://orcid.org/0000-0002-0684-8331

References
1. Kose E and Muhurcu A. The control of brushless DC motor for electric vehicle by using chaotic synchronization method. Study Inform Control 2018; 27: 403–412.
2. Ramya A, Imthiaz A and Balaji M. Hybrid self tuned fuzzy PID controller for speed control of brushless DC motor. Automatika 2016; 57: 672–679.
3. Liu K, Zhou ZQ and Hu W. A novel region refinement pulse width modulation method for torque ripple reduction of brushless DC motor. IEEE Access 2019; 7: 5333–5342.
4. Sun X, Cao J, Lei G, et al. Speed sensorless control for permanent magnet synchronous motors based on finite position set. IEEE Trans Ind Electron 2020; 67: 6089–6100.
5. Sun X, Wu M, Lei G, et al. An improved model predictive current control for PMSM drives based on current track circle. IEEE Trans Ind Electron 2020; 1–1. DOI: 10.1109/TIE.2020.2984433.
6. Cao Y, Shi T, Yan Y, et al. Braking torque control strategy for brushless DC motor with a noninductive hybrid
energy storage topology. *IEEE Trans Power Electron* 2020; 35: 8417–8428.

7. Khubalkar S, Junghare A, Aware M, et al. Modeling and control of a permanent-magnet brushless DC motor drive using a fractional order proportional-integral-derivative controller. *Turkish J Electr Eng Comput Sci* 2017; 25: 4223–4241.

8. John Prabu M, Poongodi P and Premkumar K. Fuzzy supervised online reactive-coactive neuro-fuzzy inference system-based rotor position control of brushless DC motor. *IET Power Electron* 2016; 9: 2229–2239.

9. Premkumar K and Manikandan BV. Bat algorithm optimized fuzzy PID based speed controller for brushless direct current motor. *Eng Sci Technol Int J* 2016; 19: 818–840.

10. Demirtas M. Off-line tuning of a PI speed controller for a permanent magnet brushless DC motor using DSP. *Energy Convers Manag* 2011; 52: 264–273.

11. Afrasiabi N and Yazdi MH. Sliding mode controller for DC motor speed control. *Glob J Sci Eng Technol* 2013; 2013: 45–50.

12. Rubaai A and Young P. Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives. *IEEE Trans Ind Appl* 2016; 52: 414–424.

13. Gao J and Hu Y. Direct self-control for BLDC motor drives based on three-dimensional coordinate system. *IEEE Trans Ind Electron* 2010; 57: 2836–2844.

14. Joice CS, Paranjothi SR and Kumar VJS. Digital control strategy for four quadrant operation of three phase BLDC motor with load variations. *IEEE Trans Industr Inform* 2013; 9: 974–982.

15. Premkumar K and Manikandan BV. Fuzzy PID supervised online ANFIS based speed controller for brushless DC motor. *Neurocomputing* 2015; 157: 76–90.

16. Kento K, Shin W and Shuichi O. Design of neural network PID controller based on E-FRIT. *Electr Eng Jpn* 2018; 205: 33–42.

17. Ahmed H and Rajoriya A. A hybrid of sliding mode control and fuzzy logic control using a fuzzy supervisory switched system for DC motor speed control. *Turkish J Electr Eng Comput Sci* 2017; 25: 1993–2004.

18. Badar AQH, Umre BS and Junghare AS. Reactive power control using dynamic particle swarm optimization for real power loss minimization. *Int J Electr Power Energy Syst* 2012; 41: 133–136.

19. Sun X, Jin Z, Cai Y, et al. Grey Wolf optimization algorithm based state feedback control for a bearingless permanent magnet synchronous machine. *IEEE Trans Power Electron* 2020; 35: 13631–13640.

20. Cho CN, Song YH, Lee CH, et al. Neural network based real time PID gain update algorithm for contour error reduction. *Int J Precis Eng Manuf* 2018; 19: 1619–1625.

21. Hsieh SP and Hwang TS. Dynamic modeling and neural network self-tuning PID control design for a linear motor driving platform. *IEEE Trans Electron Eng* 2010; 5: 701–707.

22. Ma L and Li D. Adaptive neural networks control using barrier lyapunov functions for DC motor system with time-varying state constrains. *Complexity* 2018; 2018: 1–9.

23. Premkumar K and Manikandan BV. Stability and performance analysis of ANFIS tuned PID based speed controller for brushless DC motor. *Curr Signal Transduct Ther* 2018; 13: 19–30.

24. Khoshdarregi MR, Tappe S and Altintas Y. Integrated five-axis trajectory shaping and contour error compensation for high-speed CNC machine tools. *IEEE/ASME Trans Mechatron* 2014; 19: 1859–1871.

25. Xia C, Li Z and Shi T. A control strategy for four-switch three-phase brushless DC motor using single current sensor. *IEEE Trans Ind Electron* 2009; 56: 2058–2066.

26. Premkumar K and Manikandan BV. Online fuzzy supervised learning of radial basis function neural network based speed controller for brushless DC motor. In: Kamalakannan C, Suresh L, Dash S, et al. (eds) *Power electronics and renewable energy systems*. New Delhi: Springer, 2015, pp. 1397–1405.

27. Vikas K, Prerna G and Mittal AP. ANN based self-tuned PID like adaptive controller design for high performance PMSM position control. *Expert Syst Appl* 2014; 41: 7995–8002.

28. Kang J, Meng W, Abraham A, et al. An adaptive PID neural network for complex nonlinear system control. *Neurocomputing* 2014; 135: 79–85.

29. Das S, Pan I, Halder K, et al. LQR based improved discrete PID controller design via optimum selection of weighting matrices using fractional order integral performance index. *Appl Math Model* 2013; 37: 4253–4268.

30. Saha S, Das S, Das S, et al. A conformal mapping based fractional order approach for sub-optimal tuning of PID controllers with guaranteed dominant pole placement. *Commun Nonlinear Sci Numer Simul* 2012; 17: 3628–3642.

31. Vinodh Kumar E and Jerome J. LQR based optimal tuning of PID controller for trajectory tracking of magnetic levitation system. *Procedia Eng* 2013; 64: 254–264.

32. Hu H, Wang T, Zhao S, et al. Speed control of brushless direct current motor using a genetic algorithm–optimized fuzzy proportional integral differential controller. *Adv Mech Eng* 2019; 11: 1–13.