International Journal of Pattern Recognition and Artificial Intelligence

A PMSM speed control strategy of LADRC based on neural network load torque estimation

--Manuscript Draft--

Manuscript Number:	IJPRAI-D-24-00801
Full Title:	A PMSM speed control strategy of LADRC based on neural network load torque estimation
Article Type:	Research Paper
Section/Category:	Patrick S P Wang (Machine Learning;Doc Analysis;Image Processing;Biometrics;Signal Processing;AI)
Keywords:	
Abstract:	This paper introduces a novel speed control method for Permanent Magnet Synchronous Motors (PMSM) utilizing Linear Active Disturbance Rejection Control (LADRC). The initial step involves collecting data on the motor's d-axis and q-axis currents, speed, and load torque during stable operation. A small-scale neural network, characterized by low computational demands, is then trained to provide a non-exact load torque estimation. Subsequently, this estimated value of the load torque is incorporated into the LADRC, resulting in a reduction of the pressure calculated by the Extended State Observer (ESO). The discrepancy between the actual and estimated load torque can be viewed as the total disturbance experienced by the system. Finally, a LADRC speed controller that utilizes an imprecise load torque estimation is constructed. To enhance the flexibility and adaptability of the method, a weight coefficient is added to the estimated load torque. The efficacy of this speed control strategy is confirmed through Matlab/Simulink simulation, demonstrating superior control performance when faced with sudden changes in load torque.

A PMSM speed control strategy of LADRC based on neural network load torque estimation

Lijie Yin, Chen Tang*, Lijun Wei
Hunan Railway Professional Technology College, No. 18 Tianxin Road, Shifeng District, Zhuzhou, Hunan, PRC.
*. Corresponding author: 372384937@qq.com

Abstract: This paper introduces a novel speed control method for Permanent Magnet Synchronous Motors (PMSM) utilizing Linear Active Disturbance Rejection Control (LADRC). The initial step involves collecting data on the motor's d-axis and q-axis currents, speed, and load torque during stable operation. A small-scale neural network, characterized by low computational demands, is then trained to provide a non-exact load torque estimation. Subsequently, this estimated value of the load torque is incorporated into the LADRC, resulting in a reduction of the pressure calculated by the Extended State Observer (ESO). The discrepancy between the actual and estimated load torque can be viewed as the total disturbance experienced by the system. Finally, a LADRC speed controller that utilizes an imprecise load torque estimation is constructed. To enhance the flexibility and adaptability of the method, a weight coefficient is added to the estimated load torque. The efficacy of this speed control strategy is confirmed through Matlab/Simulink simulation, demonstrating superior control performance when faced with sudden changes in load torque.
Key words: PMSM, ADRC/LADRC, neural network, estimated load torque

1. Introduction

Due to simple structure, compact size, high efficiency, strong dynamic performance, and other advantages, permanent magnet synchronous motor is more and more used in industrial robots, aerospace, CNC, and other fields. In practical engineering, PMSM is a nonlinear, multi-variable, highly coupled and time-varying complex system. The traditional PI speed controller is difficult to meet the increasing demand of control performance. Therefore, in PMSM speed control, how to improve the speed controller to suppress uncertainty and improve the control effect and robustness has become a hot research question [1].

To enhance control performance, numerous advanced control strategies have been adopted in the domain of speed control, like robust control [2], adaptive control [3], sliding mode control [4], etc. Active disturbance rejection control (ADRC) is an advanced control strategy that can observe and effectively control the internal and external disturbances of a system. It demonstrates an excellent control performance in the domain of PMSM control [5]. Meng, Liu, and Wang improve the extended state observer and nonlinear state error feedback by constructing novel nonlinear functions through interpolation fitting. It has a better control effect than traditional methods [6]. Wang, Bo, et al. combine ADRC with Quasi-Resonant Controllers (QRCs) to compensate for non-periodic and periodic disturbances in the PMSM speed control loop [7]. Dmitry, An, et al. propose an ADRC algorithm based on online estimation of internal and external disturbances of the system and apply it to a five-phase permanent magnet synchronous motor [8]. Tian, Wang, et al. propose a discrete-time repetitive control based ADRC for the current loop to improve the current loop control [11]. The above literature provides an in-depth discussion on ADRC/LADRC control in the field of

PMSM control, but all of them have high requirements on the observation effect of the expansion state observer (ESO), and the parameter tuning of the observer will affect the effect of the whole control.

With the continuous development of neural networks, more and more scholars apply various neural networks to permanent magnet synchronous motor control. Anh, Kien, et al. propose an artificial neural network-based magnetic field orientation control (NN-FOC) for surface-mounted permanent magnet synchronous motors (SPMSM). The neural network was trained by a modified Levenberg-Marquardt MLM algorithm and implemented for MTPA control and weak magnetic control [9]. Mao, Tang, et al. propose to optimize the control performance of PMSM through neural network and particle swarm algorithms. It first predicts the PMSM dynamic characteristics using an echo state network (ESN), and then uses a particle swarm algorithm to train the ESN output weights to solve the output weight instability problem and improve the control effect [10]. Gao, Xiong optimized the motor load system by designing a hybrid controller (RSC-EPID) combining radial basis function neural network supervisory control with expert PID control to improve the stability and speed of the system [12]. Wang, Yao, et al. proposes adaptive backstepping control to ensure system stability in the presence of time-varying parameters during motor operation and the use of radial basis function (RBF) neural networks to compensate the system for the effects of modeled load disturbances in combination with command filters to improve the control performance [13]. Ding, Xu, et al. propose a novel adaptive robust control scheme based on neural networks and robust error sign-integration methods for solving the problem of uncertain dynamics and time-varying disturbances in position in servo systems [14]. Ambrish, Madhusudan, et al. propose a current control method using hyperbolic tangent function and Levenberg-Marquardt back-propagation learning rule for artificial neural networks with better torque fluctuations than conventional controllers [15]. Velarde-Gomez, Giraldo proposes a simplified online trained Physical Information Neural Network (PINN) based on a least squares structure to identify system parameters in real-time and control them effectively [16]. All of the above literature has introduced artificial neural network algorithms into the field of PMSM control, but the complex network structure, the large amount of neural network training data, and the arithmetic power requirements needed for the operation of neural networks have somewhat restricted the popularization of neural network algorithms in PMSM control.

In this paper, a BP neural network with only 10 neurons in the hidden layer is trained using d and q-axis currents, rotational speeds, and load torque data at a steady state. The current of the d and q-axis, and rotational speeds are the network's input, and the estimated load torques as the output. The estimation will consider a known part of the system in the ADRC control method. In this way, the calculated pressure of ESO falling and the estimated effect rising. The use of small-scale BP neural networks can simplify data collection and cleaning, and also reduce the training time of the network. The small-scale neural network requires less arithmetic power during operation, which is very suitable for deployment in scenarios with limited arithmetic resources such as microcontrollers. For the problem that the small-scale BP neural network cannot accurately observe the load torque, the estimation deviation can be considered as the total disturbance of the system in ADRC control theory, to realize the effective control of the speed.

2. The model of PMSM

The analysis is simplified by making the following four assumptions for the PMSM:

(1) Neglecting factors such as core saturation, eddy current loss, and hysteresis loss;

(2) Assuming symmetrical stator winding with completely equal mutual inductance between the three phases;

(3) Considering a sinusoidal distribution of magnetic field generated by the permanent magnet;

(4) Treating stator resistance and stator inductance as constants.

Based on the above assumptions, then the d-q axis voltage equation and the electromagnetic torque equation of the motor can be written as Equation (1) (2), respectively.

$$\begin{cases} u_d = R_s i_d + L_d \dfrac{di_d}{dt} - \omega_e L_q i_q \\ u_q = R_s i_q + L_q \dfrac{di_q}{dt} + \omega_e L_d i_d + \omega_e \psi_f \end{cases} \tag{1}$$

$$T_e = \frac{3}{2} n_p \left[\psi_f i_q + \left(L_d - L_q \right) i_d i_q \right] \tag{2}$$

u_d, u_q represents the d-axis and q-axis stator voltage respectively. i_d, i_q represents the d-axis and q-axis stator current respectively. L_d, L_q represents the d-axis and q-axis stator inductance. $R_s, \psi_f, \omega_e, T_e, n_p$ stator resistance, stator flux, electric angular velocity, electromagnetic torque, number of magnetic poles.

Motor's dynamic equation is shown in Equation (3).

$$T_e - T_L = J \frac{d_\omega}{dt} + B\omega \tag{3}$$

T_L represents the load torque. J represents the moment of inertia. B represents the damping coefficient, and ω is the motor speed.

3. The PMSM speed controller based on LADRC

In the 1980s, Jingqing Han, a researcher at the Chinese Academy of Sciences, proposed a nonlinear control strategy called ADRC. The Tracking Differentiator (TD), Extended State Observer (ESO), and Nonlinear State Error Feedback (NLSEF) comprise the ADRC controller. Taking the second-order system as an example, the control block diagram is shown in Figure 1. In this figure v represents the control objective, w signifies the external disturbance, b_0 denotes the control coefficient, and y is the system output.

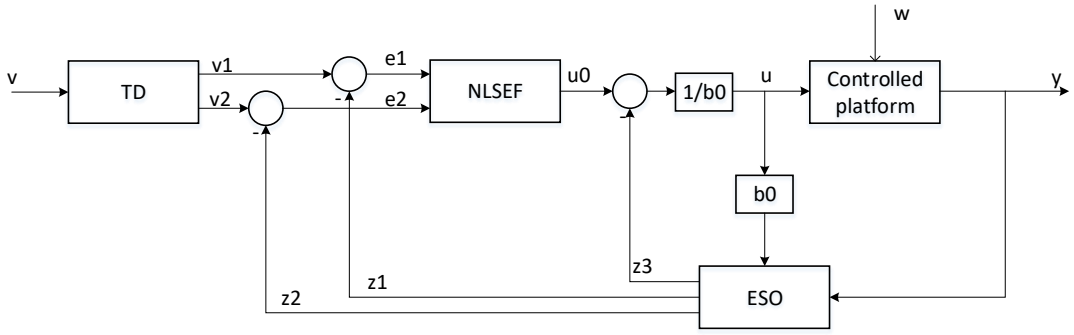


Figure 1 The block diagram of second-order ADRC

The primary goal of TD is to more effectively extract continuous and differential signals from the inherently discontinuous and noisy measurement signals. In addition, the tracking-differentiator uses the most rapid synthesis function to arrange the transition process for the control system, thus resolving the conflict between overshooting and rapidity. According to Han for the discrete system (as in Equation (4)) to deduce the most rapid integrated function $fhan(x_1, x_2, r_0, h_0)$ as in Equation (5). Where x_1, x_2 are system status, r_0, h_0 are control parameters [17].

$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k), & |u(k)| > r_0 \\ x_2(k+1) = x_2(k) - r_0 u(k), & |u(k)| \le r_0 \end{cases} \tag{4}$$

$$\begin{cases} d = r_0 h_0^2 \\ a_0 = h_0 x_2 \\ y = x_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + sign(y)(a_1 - d)/2 \\ s_y = [sign(y+d) - sign(y-d)]/2 \\ a = (a_0 + y - a_2)s_y + a_2 \\ s_a = [sign(a+d) - sign(a-d)]/2 \\ fhan = -r[a/d - sign(a)]s_a - r_0 sign(a) \end{cases} \tag{5}$$

The discrete fastest feedback system can be established by using Equation (4), as shown in Equation (6). Where x_1 quickly tracks the input signal v without overshoot, and x_2 is an approximate differential of v.

$$\begin{cases} fh = fhan(x_1(k) - v(k), x_2(k), r_0, h_0) \\ x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+1) = x_2(k) + hfh \end{cases} \tag{6}$$

The ESO is the core of ADRC. The total perturbation is expanded into a new state variable, and then all states containing the original state variables of the system with the perturbation are observed using the system inputs and outputs. As an example, a second-order system is provided with second-order objects:

$$\ddot{y} = \mathrm{f} + bu \tag{7}$$

Where f is the total disturbance that combines the external disturbance and the internal disturbance of the system (the deviation of the mathematical model of the system from the actual system). Let $x_1 = y, x_2 = \dot{y}$, the equation of state is obtained:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f + bu \\ y = x_1 \end{cases} \tag{8}$$

Treating f as a new unknown state variable is equivalent to expanding a new state in the original system, which becomes:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = \dot{f} \\ y = x_1 \end{cases} \tag{9}$$

The nonlinear state observer can be built for Equation (9):

$$\begin{cases} \varepsilon_1 = z_1 - y_1 \\ \dot{z}_1 = z_2 - \beta_{01}\varepsilon_1 \\ \dot{z}_2 = z_3 - \beta_{02} fal(\varepsilon_1, \frac{1}{2}, \delta) + bu \\ \dot{z}_3 = -\beta_{03} fal(\varepsilon_1, \frac{1}{4}, \delta) \end{cases} \tag{10}$$

Where $fal(x, a, \delta)$ is a nonlinear function:

$$fal(x, a, \delta) = \begin{cases} \dfrac{x}{\delta^{(1-a)}}, & |x| \le \delta \\ sign(x)|x|^a, & |x| > \delta \end{cases} \tag{11}$$

NLSEF can combine the three signals of error signal, error differential, and error integral to synthesize either a linear combination similar to PID control, or construct a nonlinear controller using the fal function and $fhan$ function. Taking the fal function as an example, there can be nonlinear combinations:

$$\begin{cases} e_1 = v_1 - z_1 \\ e_2 = v_2 - z_2 \\ u_0 = \beta_1 fal(e_1, a_1, \delta) + \beta_2 fal(e_2, a_2, \delta) \end{cases} \tag{12}$$

where v_1, v_2 are the target values, $z_1 = \widehat{x_1}, z_2 = \widehat{x_2}$. Then the perturbation compensation can be formed into a control quantity:

$$u = \frac{u_0 - z_3}{b_0} \tag{13}$$

In the above ADRC algorithm, $r_0, \beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, a_1, a_2, \delta, b_0$ are the controller parameters, which are difficult to be rectified, and this affects the promotion of the application of ADRC to a certain extent. Professor Zhiqiang Gao linearized the ADRC and proposed the LADRC. LADRC greatly simplifies ADRC, which is of great significance for its engineering application promotion [18][19].

By applying LADRC to PMSM speed control, the control equations can be established as in Equation (14) without the aid of any mathematical model.

$$\dot{\omega} = f + b_0 u \tag{14}$$

Where ω, f, b_0, u is motor mechanical rotational speed, system disturbance, control coefficient, controller input respectively. In motor speed control, the controller input is electromagnetic torque. When $i_d = 0$ control method is used in PMSM, the torque equation is shown as Equation (15).

$$T_e = \Psi i_q \tag{15}$$

Where Ψ, T_e is flux linkage, electromagnetic torque respectively. In the process of speed control, the flux change speed is much slower than the current change, so it can be approximated that if the flux is constant, the electromagnetic torque is only related to i_q. So Equation (14) can be transformed into Equation (16).

$$\dot{\omega} = f + b_0 i_q \tag{16}$$

In the process of converting from Equation (14) to Equation (16), there are many errors, such as the error caused by ignoring Ψ. These errors are aggregated into f, constituting the total disturbance of the system.

From Equation (14), state equations can be constructed as Equation (17) and extended observer equations as Equation (18).

$$\begin{cases} x_1 = \omega \\ \dot{x}_1 = x_2 - \alpha T_L + b_0 i_q \\ \dot{x}_2 = \dot{f} \end{cases} \tag{17}$$

$$\begin{cases} \dot{z}_1 = z_2 + b_0 i_q + \beta_1 (z_1 - \omega) \\ \dot{z}_2 = \beta_2 (z_1 - \omega) \end{cases} \tag{18}$$

Where z_1, z_2 represent the estimates of x_1, x_2 respectively, $\widehat{x_1}, \widehat{x_2}$. β_1, β_2 are observer parameters respectively. The design control rate is shown in Equation (19):

$$i_q = \frac{1}{b_0} (l_1 (\omega^* - z_1) - z_2) \tag{19}$$

Where ω^*, l_1 are aim speed, controller parameter respectively.

Substitute Equation (19) into Equation (16):

$$\dot{\omega} = f + l_1 (w^* - z_1) - z_2 \tag{20}$$

According to Equation (17), z_2 is an estimate of f, and when ESO achieves the expected estimation effect, z_2 will cancel out with f. And because z_1 is the estimated speed value, Equation (20) is further simplified into speed error multiplied by the coefficient. It similar to proportional control form.

When PI controller is used in the current loop and LADRC controller is used in the speed loop, the control block diagram is shown in Figure 2.

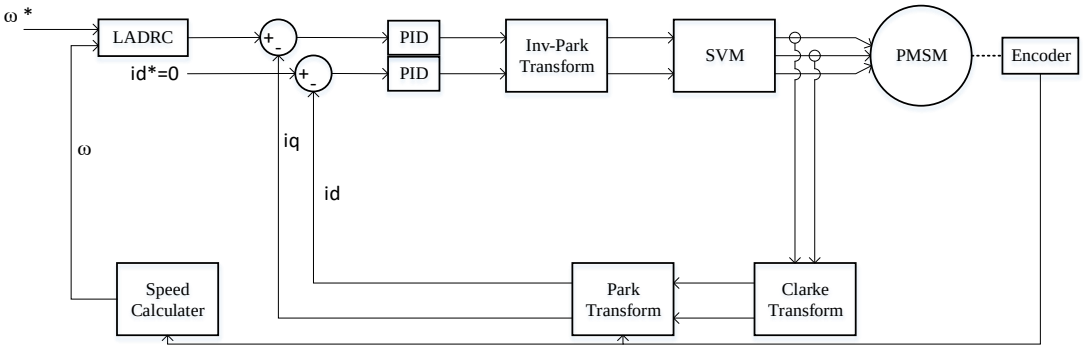


Figure 1 The speed control block diagram of LADRC

4. Load torque estimation based on BP neural network

There are many types of neural networks nowadays. Each type has its own suitable application scenarios and has different requirements for computing power. BP neural network is a type of network with a simple structure and relatively low computing power requirements, which can run well on microcontrollers.

The BP neural network comprises multiple layers of neuron nodes, mainly encompassing the input layer, hidden layer, and output layer. Each neuron node is linked to all neurons in the previous layer through weighted summation for signal transmission, which are subsequently subjected to nonlinear transformation by the activation function. This multilayered structure empowers the BP neural network to handle intricate nonlinear problems. As depicted in Figure 3, in this paper, the d-axis and q-axis currents i_d and i_q, along with the motor speed ω, are regarded as the neural network inputs. They enter the hidden layer after traversing the input layer and reach the output layer upon calculation, where the estimated load torque T_L is output based on these three inputs.

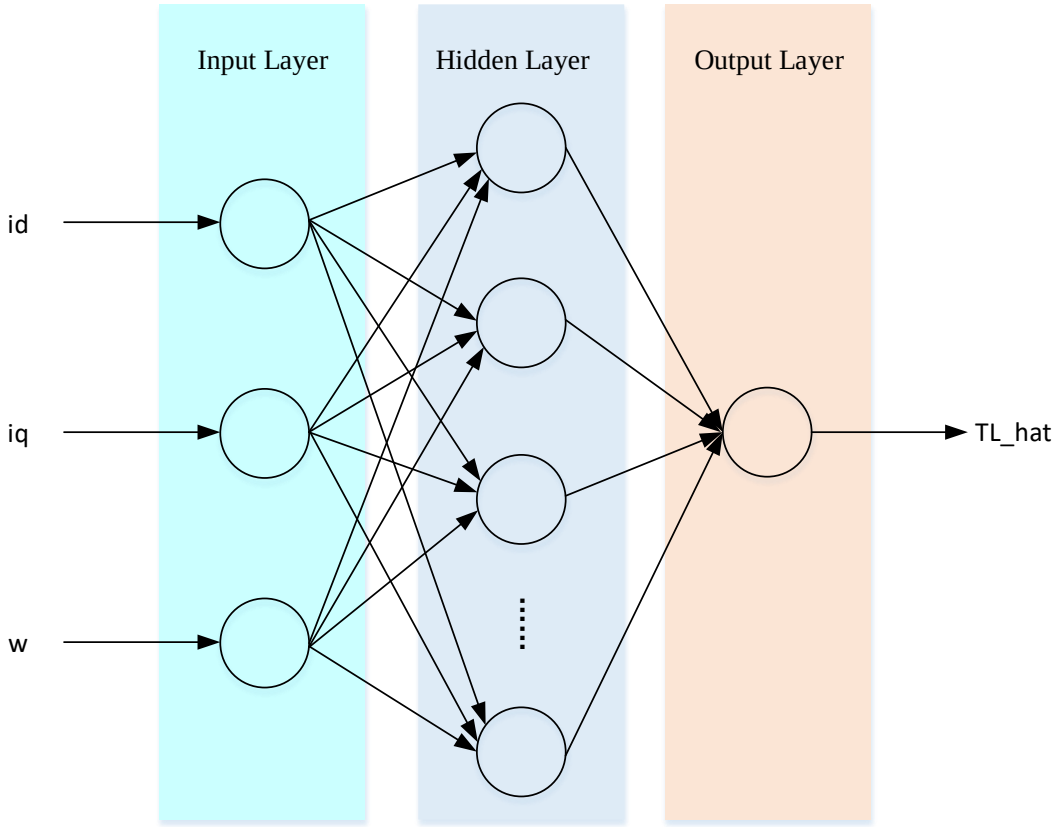


Figure 2 BP neural network basic structure diagram

The training of a BP neural network consists primarily of two phases: forward propagation and backward propagation. Forward propagation involves the movement of input data through the network, starting from the input layer, passing through hidden layers where it undergoes weighted and non-linear transformations, and ultimately reaching the output layer to generate results. This flow of data from input to output is termed forward propagation. Conversely, backward propagation occurs when there is a discrepancy between the actual output and expected results; this process entails transmitting errors back from the output layer to both hidden layers and the input layer. During backward propagation, each neuron's gradient (the partial derivative concerning weight and bias relative to error) is computed iteratively for each layer using error values in conjunction with the chain rule. Based on these gradients, adjustments are made to weights and biases within the network aimed at minimizing errors. This iterative process continues until the network's outputs achieve an acceptable level of accuracy.

In accordance with the motor's dynamic equation, as exemplified by Equation (3), the rotational speed of the motor is interlinked with the load torque, electromagnetic torque, moment of inertia, as well as damping coefficient. When the mechanical system remains unchanged, the moment of inertia and damping coefficient can be approximately considered constant. At this time, the only factors affecting the motor's rotational speed are the electromagnetic torque and the load torque. Combining the electromagnetic torque equation of the motor, such as Equation (2), when the $i_d = 0$ control strategy is used, the electromagnetic torque is related to the flux, q-axis current i_q, and the number of pole pairs. Therefore, the relationship between the motor's rotational speed, i_q, and the load torque can be modeled using a BP neural network with its nonlinear modeling and generalization capabilities. By collecting data and training the BP neural network, the goal is to

achieve the input of the motor's rotational speed and i_q and the output of the load torque.

5. The speed control strategy of LADRC based on the BP neural network

After training, the BP neural network can estimate the current load torque by inputting the current motor speed and i_q. However, in real-world environments, the changes in L_q, L_d, etc. caused by temperature changes will result in a difference between the actual load torque and the estimated value. Adding more environmental information to the training data can effectively improve the accuracy of the estimator. But it will also greatly increase the size of the neural network, increasing the difficulty of neural network training and the computational complexity when using the trained network.

Because there are estimation errors, it is not suitable to directly incorporate the estimated load torque into the speed control through feedforward compensation. Especially when the estimated output fluctuates, the feedforward compensation may have a negative effect on the speed control. However, it is very suitable to incorporate the estimated load torque into the LADRC for speed control. The estimation error can be considered as a disturbance of the system, and the introduction of the estimated load torque will reduce the pressure on the expanded state observer, making the speed control have better effects.

By increasing the estimated load torque $\hat{T}_L$ from Equation (16), the following equation can be obtained:

$$\dot{\omega} = f - T_L + b_0 i_q \tag{21}$$

In order to increase the flexibility of the controller and adapt to the requirements of different application scenarios, the weight coefficient α is added to the estimated load torque on the basis of Equation (21) to obtain Equation (22):

$$\dot{\omega} = f - \alpha T_L + b_0 i_q \tag{22}$$

State equations and observation equations are constructed against Equation (23):

$$\begin{cases} x_1 = \omega \\ \dot{x}_1 = x_2 - \alpha T_L + b_0 i_q \\ \dot{x}_2 = \dot{f} \end{cases} \tag{23}$$

$$\begin{cases} \dot{z}_1 = z_2 - \alpha T_L + b_0 i_q + \beta_1 (z_1 - \omega) \\ \dot{z}_2 = \beta_2 (z_1 - \omega) \end{cases} \tag{24}$$

Continue using the control rate shown in Equation (19), which constitutes a LADRC speed control system based on BP neural network load torque estimation, as shown in Figure 4. Compare the new control strategy (Equation (22)) with the traditional first-order LADRC speed control strategy (Equation (16)). The former includes an additional $\alpha\hat{T}_L$ term, while the dynamic equation of the permanent magnet synchronous motor (Equation (3)) indicates that the speed is related to the

load torque. Therefore, in the traditional strategy, the load torque is also considered a system disturbance, which is estimated by extending the state observer. Within the novel strategy, the load torque is estimated through the BP neural network and then incorporated into the speed control process. This approach serves to alleviate the observation burden of the ESO and enhance the observation efficacy, thereby attaining a more favorable control performance.

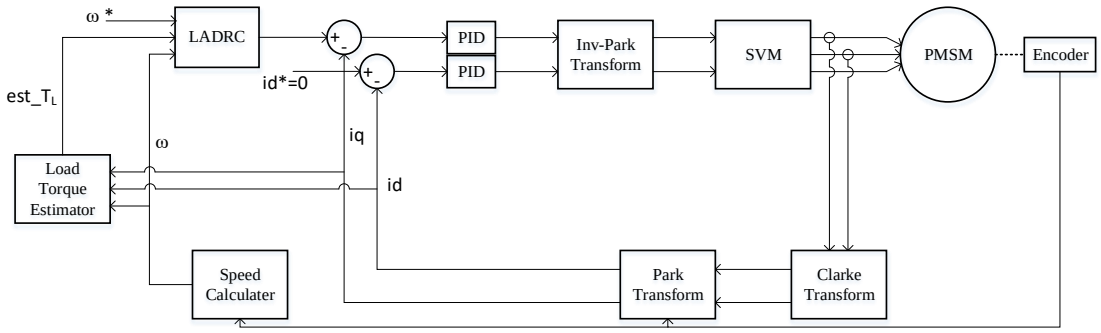


Figure 4 The diagram of speed control strategy of LADRC based on BP neural network load torque estimation

6. Simulations

To validate the efficacy of the speed control strategy of LADRC based on BP neural network load torque estimation, a simulation platform was constructed on the MATLAB/Simulink platform. The parameters of the PMSM were selected as presented in Table 1.

Table 1 Testing motor parameters

Parameters	Value	Unit
Rated voltage	300	V
Rated torque	6	Nm
Number of pole pairs	4	
Stator resistance	0.62	Ohm
Stator inductance	0.002075	H

The current loop PID adopts parallel structure, and the controller parameters directly use the stator resistance and inductance of the motor to calculate the K_p and K_i parameters.

$$\begin{cases} K_p = L_s * \omega_c \\ K_i = R_s * \omega_c \end{cases} \tag{25}$$

The speed loop is also controlled by PID，and $K_p = 0.08$，$K_i = 1$，$K_d = 0$. When the load torque is 0 and the target speed is set to 2700rpm (since there is 3000rpm training data when training BP neural network, 2700rpm is used as the target speed in simulation tests), the speed waveform is shown in Figure 5. It can be seen from the waveform diagram that the maximum speed reaches 2945.1rpm and the overshoot is 9%. At 0.069s, the speed reaches 2835.19rpm, basically entering the range of 5%, and then the speed gradually tends to 2700rpm. At 0.19s, the rotational speed reached 2727.02rpm, and the rotational speed basically reached the target value and tended to be stable.

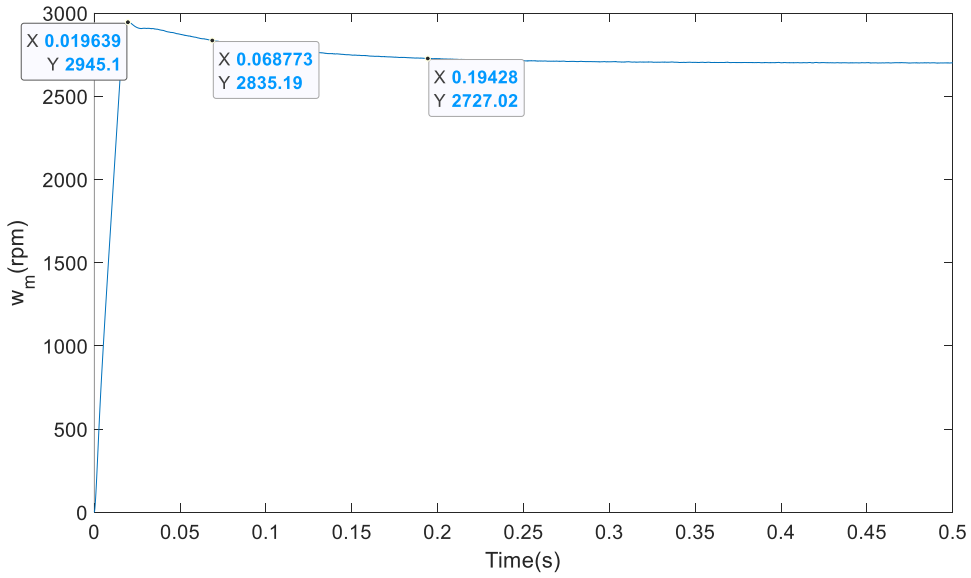


Figure 5 Control effect diagram of the PID speed controller when no load

At the beginning of setting, the motor has no load, and at 0.3s, the load torque increases by 3Nm. The speed waveform is shown in Figure 6. In the figure, the speed decreased significantly at about 0.3s. At 0.3024s, the speed decreased to the lowest value of 2617.43rpm, at which time the speed decreased by 3%. At 0.3726s, the speed returned to 2673.11rpm, and the adjustment time for the speed to return to within 1% of the target speed was about 0.07s.

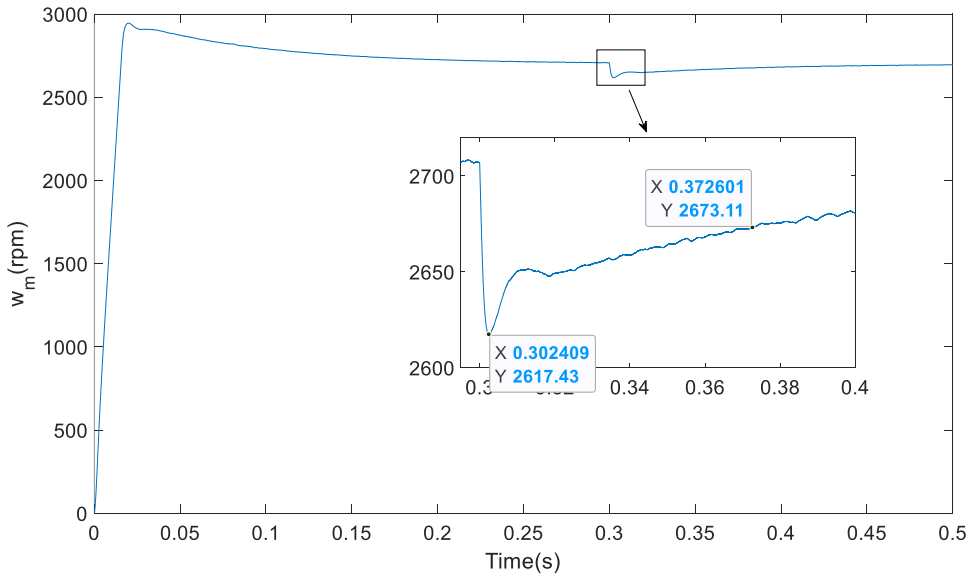


Figure 6 Control effect diagram of the PID speed controller when load torque mutation

The speed controller based on LADRC has a better control effect than the PI speed controller, and its simulation waveform is shown in Figure 7. At 0.0159s, the maximum speed reaches 2722.59rpm, and the overshoot is only 0.84%. If the error of 5% is used as the adjustment time, the adjustment time is only 0.0159s. At about 0.019s, the speed reaches 2696.86rpm, and the speed error is only 0.12%. Compared with PI control, the speed control effect has been greatly improved.

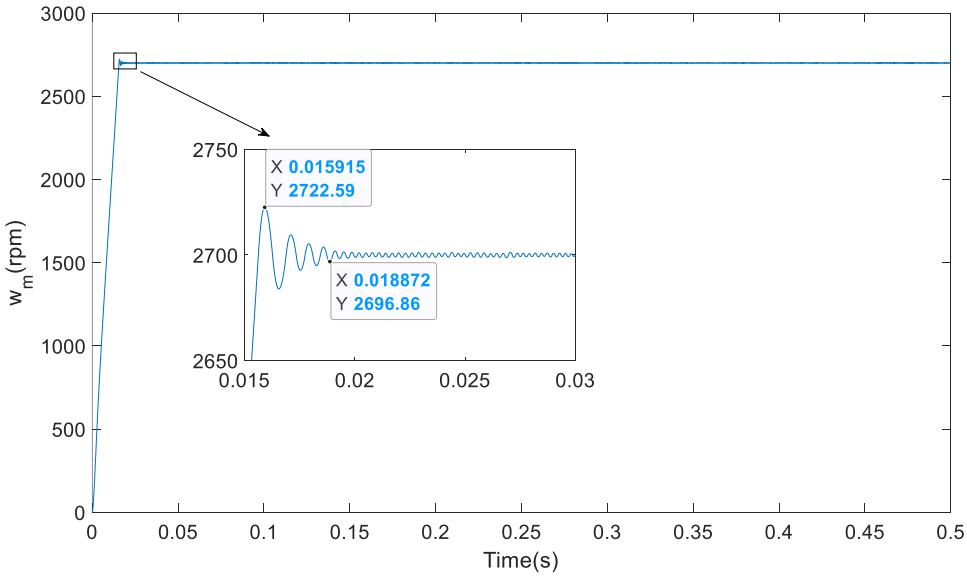


Figure 7 Control effect diagram of the LADRC speed controller when no load

The motor is set to start with no load torque and a sudden increase of 3Nm load at 0.3s, and the speed waveform using the LADRC speed controller is shown in Figure 8. Without local amplification of the speed waveform, the effect of the sudden increase of 3Nm load torque at 0.3s on the speed can hardly be seen. The local magnification of 0.29s~0.31s shows that the speed suddenly drops at 0.3s due to the sudden increase of load torque, and the lowest speed reaches 2686.45rpm. In other words, under the control of LADRC speed controller, the sudden increase of 3Nm load torque only changes the speed by 0.5%, and the control effect is also significantly improved than that of PI controller.

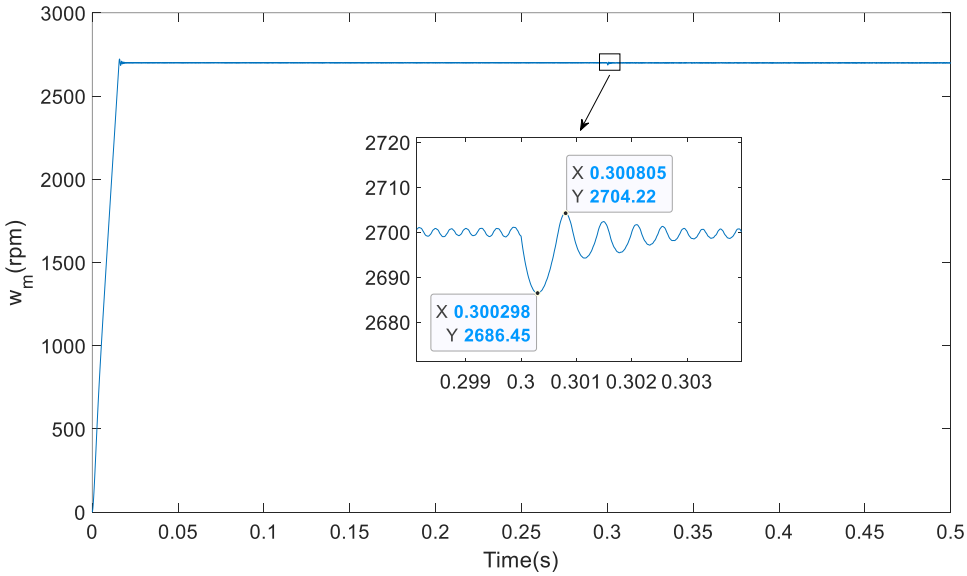
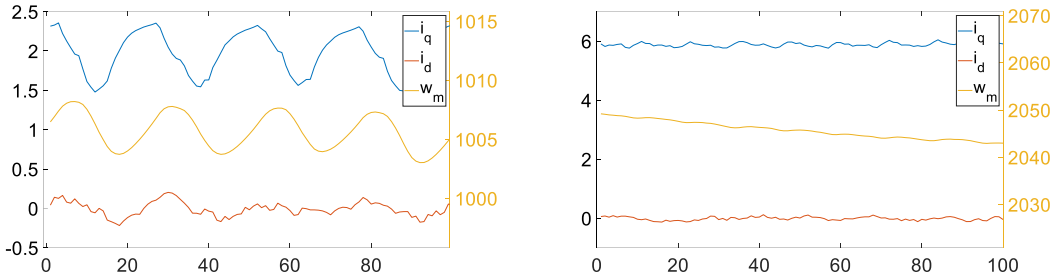


Figure 8 Control effect diagram of the LADRC speed controller when load mutation

The LADRC speed controller based on neural network load torque estimation can reduce the pressure of the expanded state observer in LADRC to achieve better speed control effects by being able to estimate the load torque. The training data for the BP neural network is as follows: the speed is 100, 200, 500, 800, 1000, 1200, 1500, 1800, 2000, 2100, 2300, 2500, 2800, and 3000 rpm, the

load torque is 0, 1, 2, and 3 Nm, the speed, load torque, and d, q axis current data are collected, a total of 5200 sets of data. The above data was obtained in a system with a current loop and a speed loop using PI control, so there are fluctuations and errors in the data. Figure 9 shows the visualized training data. Figure 9(a) presents the training data under the conditions of a rotational speed of 1000 rpm and a load torque of 1 Nm. It can be observed that the speed data exhibits fluctuations and does not precisely match the target value. Meanwhile, Figure 9(b) illustrates the training data when the speed is 2000 rpm and the load torque is 3 Nm. In this case, the value of i_q is approximately 6A, and likewise, the speed also contains error.



(a) speed 1000rpm，load torque 1Nm　　(b) speed 2000rpm，load torque 3Nm

Figure 9 Plot of the training data for visualization

The BP neural network is structured with 3 input neurons, 10 neurons within the hidden layer, and 1 neuron in the output layer. The activation function of the hidden layer neurons is sigmoid, and the activation function of the output layer neuron is a linear function. The training is performed using Bayesian regularization in the nftool tool of Matlab. After 868 training rounds, the minimum gradient reached 9.14e-8, and the training was stopped. The trained model was imported into Simulink, and the i_d, i_q and speed data were connected. The estimated load torque output was connected to the speed controller of the LADRC. The estimated load torque waveform is shown in Figure 10, starting from t=0, the estimator estimates the load torque to be 3Nm. At this time, the motor is in an acceleration process, i_q is large and the speed value is small, which causes the model to be unable to effectively track the load torque trajectory. After the load torque is suddenly increased to 3Nm at t=0.3s, the load torque estimator accurately estimates the load torque after only 0.0002s. In the local enlarged graph, the estimated load torque fluctuates between 2Nm and 3Nm, and this fluctuation value can be regarded as the overall disturbance of the system.

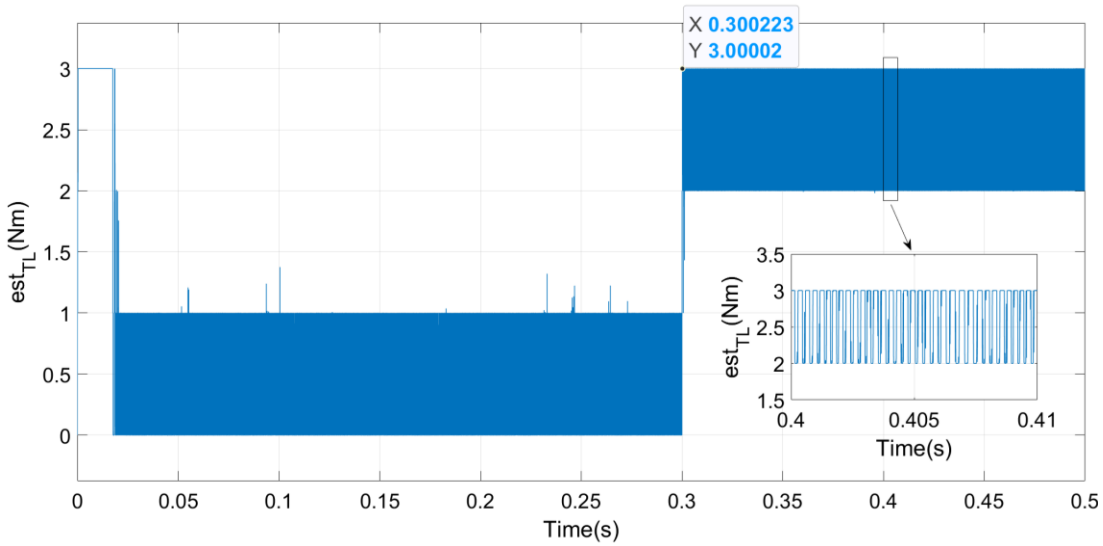


Figure 10 The waveform of the neural network estimates the load torque

The speed controller effect can be further improved by adding the output of the neural network load torque estimator to the LADRC speed controller, whose waveform is shown in Figure 11. The motor is set to start with no load torque and a sudden increase of 3Nm load at 0.3s. The local magnification of 0.295s~0.305s shows that the speed suddenly drops at 0.3s due to the sudden increase of load torque, and reaches the lowest speed to 2693.18rpm at 0.3002s, and then the speed starts to rise again. In the case of a sudden increase of load torque of 3Nm, only 0.25% of the speed fluctuation is caused, and the control effect is improved compared with the ordinary LADRC controller.

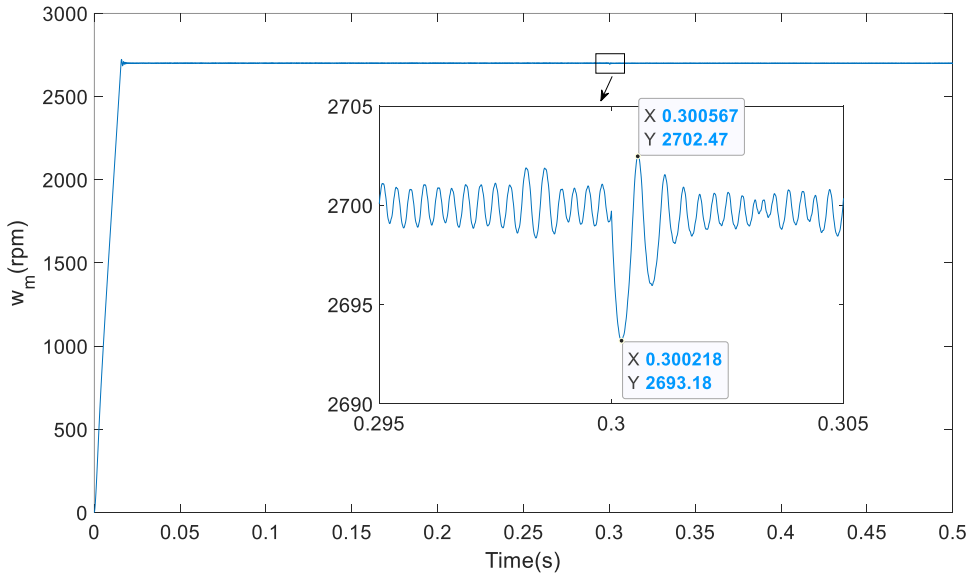


Figure 11 speed waveform of LADRC based on BP neural network load torque estimation when load mutation

When the load torque changes from large to small, it also has a similar performance. The speed control strategy of LADRC based on BP neural network load torque estimation has smaller speed fluctuations when the torque is mutated, as shown in Figure 12.

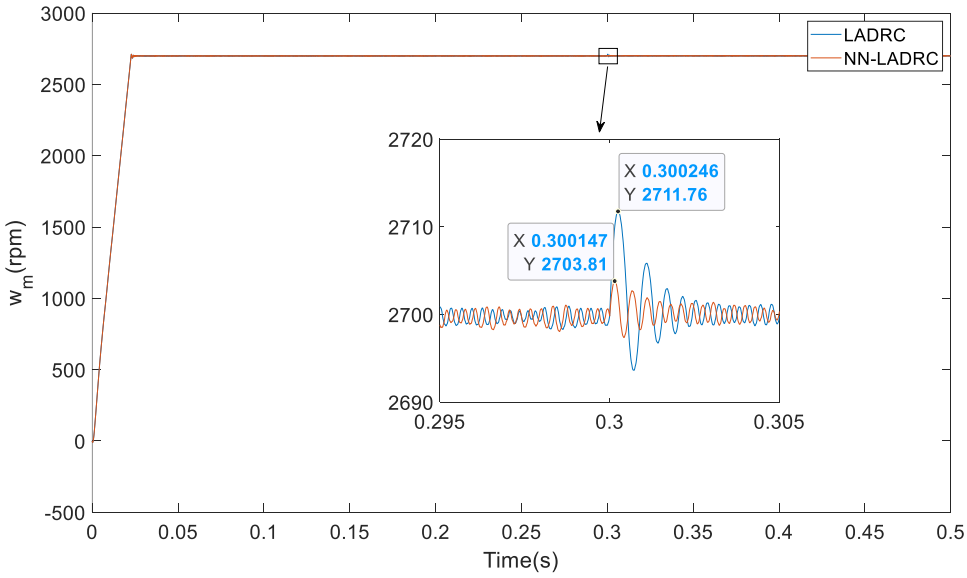
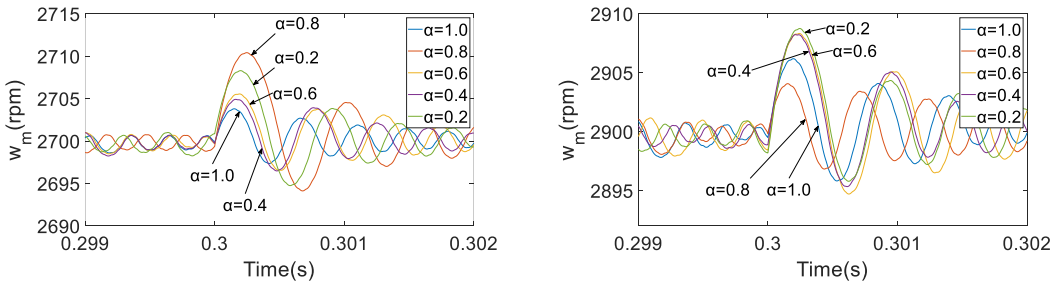


Figure 12 speed waveform of LADRC & NN-LADRC when load dropping

At different target speeds, different coefficients α will result in different control effects. The simulation results at 2700rpm and 2900rpm with different weighting coefficients are shown in Figure 13. In this figure, only the response to load torque changing from 3Nm to 0Nm from 0.299s to 0.302s is of interest. At 2700rpm, the control effect of $\alpha = 1.0$ is the best, while the control effect of $\alpha = 0.8$ is the worst. However, at 2900rpm, the control performance of $\alpha = 0.8$ is the best. Therefore, the weighting coefficient α can be adjusted according to the different needs of the target speed to obtain the best control effect.



a Comparison of different α at 2700rpm b Comparison of different α at 2900rpm

Figure 13 Comparison of the speed control effect of different α

7. Conclusion

This paper proposes a speed control strategy for PMSM that utilizes a compact BP neural network to estimate load torque and integrates this estimation with LADRC. Based on the traditional LADRC speed controller, the BP neural network is used to estimate the load torque, and the estimated load torque is directly applied to the linear active disturbance rejection speed controller. The introduction of load torque reduces the observation pressure of the expanding observer and improves the dynamic performance of the speed controller. The simulation results on the MATLAB/Simulink platform indicate that the proposed algorithm can enhance the dynamic performance beyond what is achieved by the existing LADRC speed controller. When the load torque mutation occurs, the controller can restore the motor to the set speed faster, reduce the motor speed fluctuation caused by the load torque change, and make the motor better applied to the scene that is sensitive to the speed fluctuation caused by the torque change.

Funding: This research was funded by Hunan Railway Professional Technology College Electromagnetic Molecular Acceleration System and Equipment Development Innovation Team, grant number KYTD202402; The Scientific Research Projects of the Hunan Provincial Department of Education (No. 23C0806).

References

[1] Ma, Y.; Li, Y. Active Disturbance Compensation Based Robust Control for Speed Regulation System of Permanent Magnet Synchronous Motor [J]. Applied Sciences.2020, 10(2):709.

[2] Mohamed YA-RI. Design and implementation of a robust current-control scheme for a PMSM vector drive with a simple adaptive disturbance observer[J]. IEEE Trans Ind Electron, 2007, 54(4):1981–1988.

[3] Choi HH, Vu NT-T, Jung J-W. Digital implementation of an adaptive speed regulator for a PMSM[J]. IEEE Trans Power Electron,2011,26(1):3–8.

[4] Zhang X, Sun L, Zhan K, Sun L. Nonlinear speed control for PMSM system using sliding-mode control and disturbance compensation techniques[J]. IEEE Trans Power Electron, 2013,28(3):1358–1365.

[5] B. Guo, S. Bacha and M. Alamir. A review on ADRC based PMSM control designs[C]. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. 2017:1747-1753.

[6] Meng Y B, Liu B Y, Wang L C. Speed control of PMSM based on an optimized ADRC controller[J]. Mathematical Problems in Engineering, 2019(1): 1074702.

[7] Wang, Bo, Tian, Minghe, Yu, Yong, Dong, Qinghua, Xu, Dianguo. Enhanced ADRC With Quasi-Resonant Control for PMSM Speed Regulation Considering Aperiodic and Periodic Disturbances[J]. IEEE TRANSACTIONS ON TRANSPORTATION ELECTRIFICATION, 2022,8(3):3568-3577.

[8] Semenov Dmitry; An QunTao; Sun Li; Mirzaeva Galina. Nonlinear speed control for Five-phase PMSM in Electric Vehicles[C]. PROCEEDINGS OF THE 2016 AUSTRALASIAN UNIVERSITIES POWER ENGINEERING CONFERENCE (AUPEC),2016.

[9] Ho Pham Huy Anh; Cao Van Kien; Tran Thien Huan; Pham Quoc Khanh. Advanced Speed Control of PMSM Motor Using Neural FOC Method[C]. PROCEEDINGS OF 2018 4TH INTERNATIONAL CONFERENCE ON GREEN TECHNOLOGY AND SUSTAINABLE DEVELOPMENT (GTSD), 2018.

[10] Mao, Hubo; Tang, Xiaoming; Tang, Hao. Speed control of PMSM based on neural network model predictive control[J]. TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT AND CONTROL,2022, 44(14):2781-2794.

[11] Tian, Minghe; Wang, Bo; Yu, Yong; Dong, Qinghua; Xu, Dianguo. Discrete-Time Repetitive Control-Based ADRC for Current Loop Disturbances Suppression of PMSM Drives[J]. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS,2022, 18(5):3138-3149.

[12] Gao, Hongliang; Xiong, Lang. Research on a hybrid controller combining RBF neural network supervisory control and expert PID in motor load system control[J]. ADVANCES IN MECHANICAL ENGINEERING, 2022, 14(7).

[13] Wang, Xiuping; Wang, Yiming; Yao, Shunyu; Qu, Chunyu; Wang, Hai. Adaptive backstepping control of primary permanent magnet linear motor via radial basis function neural network and command filter[J]. COMPUTERS & ELECTRICAL ENGINEERING, 2023, 109.

[14] Ding, Runze; Ding, Chenyang; Xu, Yunlang; Liu, Weike; Yang, Xiaofeng. Neural network-based robust integral error sign control for servo motor systems with enhanced disturbance rejection

performance[J]. ISA TRANSACTIONS, 2022, 129.
[15] Devanshu, Ambrish; Singh, Madhusudan; Kumar, Narendra. Artificial neural network-based current control of field oriented controlled induction motor drive[J]. ELECTRICAL ENGINEERING, 2021, 103(2):1093-1104.
[16] Velarde-Gomez S, Giraldo E. Real-Time Identification and Nonlinear Control of a Permanent-Magnet Synchronous Motor Based on a Physics-Informed Neural Network and Exact Feedback Linearization[J]. Information, 2024; 15(9):577.
[17] J. Han. From PID to Active Disturbance Rejection Control[J]. IEEE Transactions on Industrial Electronics, 2009,56(3):900-906.
[18] Z. Gao. Scaling and bandwidth-parameterization based controller tuning[C]. Acc. 2003, 4: 989-4.
[19] Z. Gao. From Linear to Nonlinear Control Means: A Practical Progression[J]. ISA Transactions, 2002,41(2):177-189.

Lijie Yin received his BSc degree in Electronic Information Engineering from Hunan University of Technology and his MSc degree in Electronics and Communication Engineering from Xiangtan University. Currently, he is a lecturer at Hunan Railway Professional Technology College, specializing in intelligent control & motor control.

Chen Tang received his BSc degrees in Automation from South-Central University for Nationalities, China and MSc degree in Control Science from Hunan University of Science and Technology, China. Currently, he is a lecturer at Hunan Railway Professional Technical College. His research interests include machine vision and deep learning.

Lijun Wei received his BSc degree in Electronic Information Engineering from Hunan University of Arts and Science and his MSc degree in Earth Exploration and Information Technology, Central South University. Currently, he is an associate professor at Hunan Railway Professional Technology College, specializing in application of Electronic Technology and Sensor Technology.