A Starter Performance Testing System Based of PIDNN Controller

Guodong Zhou [1, a], Dunli Zhang [1, b]

[1] School of Intelligent Manufacturing, Hunan Open University, Changsha, 410004, China.

[a] zgd79@qq.com, [b] 11506043@qq.com

Abstract. Starter performance tester requires high precision and fast speed with a complex nonlinear structure and strong interference. The traditional PID controller always brings the bad effect with larger current and torque shocks. The complex intelligent algorithm is difficult to be realized in practical engineering for its trouble of parameter tuning. A new PID neural network(PIDNN) controller is designed which integrated the advantages of PID and intelligent controller. PIDNN has simple structure as PID and self-learning ability as NN to optimize parameters. The experiments verified the precision and efficiency of the test system.

Keywords: DC motor; starter; PID; neural network.

1. Introduction

The automotive starting motor testing system has complex nonlinear and strong interference characteristics, high testing accuracy, fast testing speed, and high requirements for the controller[1]. At present, the main algorithm used in engineering practice is the PID algorithm. Practice has shown that PID has poor performance on nonlinear systems, with large output fluctuations and easy occurrence of large current and torque shocks, resulting in poor performance.

In recent years, intelligent algorithms have been applied to motor testing systems, including Active Disturbance Rejection Control (ADRC) algorithm[2], Sliding Membrane Variable Structure Theory[3], Fuzzy Control[4], and so on. However, most intelligent algorithms have complex structures, numerous parameters, and unclear physical meanings, making tuning and optimization difficult. Although the above methods have achieved good results in the laboratory, they are not easy to apply in engineering practice.

PID neural network (PIDNN) is a structurally simple and algorithmic efficient control method used in nonlinear system control[5]. It originated from traditional PID control algorithms, inheriting the advantages of PID control, such as not requiring precise mathematical models of the controlled system, simple structure, and efficient algorithms. It also combines the nonlinear mapping characteristics and self-learning ability of neural networks[6,7].

The article analyzes the characteristics of the mathematical model of the automotive starter testing system and applies the PIDNN algorithm to the starter motor testing system. The PID element is used as the hidden layer neuron of the neural network, which simplifies the structural design of the neural network and clarifies the physical meaning of the neuron. Then, by utilizing the nonlinear mapping law and online optimization ability of the neural network, the weight parameters of the PID are automatically adjusted.

2. Model Analysis Of Starting Motor Testing System

Figure 1 shows the schematic diagram of performance testing for automotive starters. Its control task is to adjust the output speed and simulated load torque of the starting motor through two different adjustment circuits, and then determine whether the output current is qualified. The PIDNN controller completes the dual control decoupling output of simulated load torque and speed.

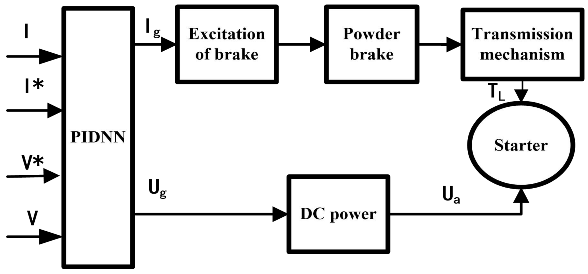


Fig. 1 Diagram of starter testing system

The state equation[8] of the testing system is formula(1).

$$\begin{cases} L_a\dot{i}_a = U_a - i_a r - C_e v \\ J\dot{v} = T - T_L - T_d = C_T i_a - (k i_g + d) - T_d \end{cases} \tag{1}$$

In the formula, U_a is the rotor voltage of the motor; L_a is the rotor inductance; r is the internal resistance value of the rotor; i_a is the rotor current value; C_e is the electromotive force coefficient of the motor; J is the moment of inertia of the motor; v is the output speed of the rotor; Motor torque $T = C_T i_a$, where C_T is the torque coefficient; T_L is the motor load torque which is the output torque of the magnetic powder brake. The relationship between the output torque T_L of the magnetic powder brake and the input signal i_g is formula (2).

$$T_d = k i_g + d \tag{2}$$

In the formula, k and d are the values of the linear relationship coefficient which can be obtained through regression of experimental data; T_d is the total sum of unknown disturbance torques in the system.

In summary, in this system, U_a and i_g are the output quantities, and i_a and v are the input quantities.

3. PIDNN Control Algorithm

According to the analysis of the starting motor testing system model in the previous text, the system should have a structure of two inputs and two outputs. Therefore, i_a and v are selected as the state variables of the testing system, U_a and i_g are correspondingly selected as the control variables of the testing system.

The network topology of the PIDNN controller is designed as shown in Figure 2.

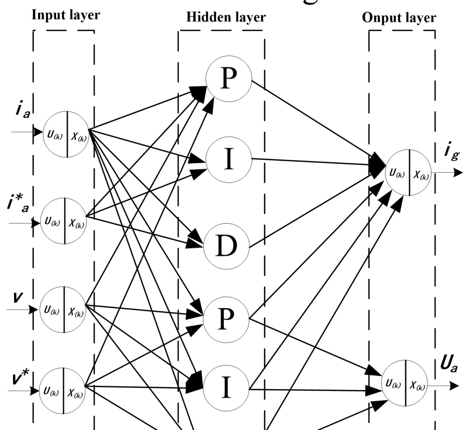


Fig. 2 Network structure of PIDNN algorithm

The PIDNN controller for the starting motor testing system is designed as a typical three-layer neural network. The input layer of the first layer contains four neurons, namely the target values of i_a and v, and the test sensor return values i_a^* and v_*. Its task is to convert the input physical quantity to the input value of the neural network, usually through normalization processing. In this system, all four input values are already normalized, so there is no need to calculate; The second layer is the improved hidden layer, which has 6 PID neurons (two for each PID), and its task is to implement the PID algorithm; The output layer of the third layer contains two neurons, whose task is to achieve the output of control variables. There are connections between each unit in the input layer and each unit in the hidden layer, and the same goes for the hidden layer to the output layer. Figure 2 shows clearly, only the connections of the first node are drawn, and the remaining nodes are omitted.

3.1 Forward Path Algorithm

The task of the forward path of a neural network is to transmit signals from the input layer to the hidden layer, and then from the hidden layer to the output layer. In this system, its task is to receive the target adjustment value, calculate it through a neural network, and output the U_a and i_g required for testing the starter. The algorithms[9] for each layer are as follows.

3.1.1 Input layer algorithm

$$x_i(k) = u_i(k) \tag{3}$$

In the formula, i is the sequence number of the input layer neurons, with values ranging from 1 to 4. k is the sequence number, $u_i(k)$ is the original input value of each neuron in the input layer, and $x_i(k)$ is the conversion output value of each neuron in the input layer.

3.1.2 Hidden layer algorithm

The hidden layer contains 2 proportional neurons P (numbered 1 and 4), 2 integral neurons (numbered 2 and 5), and 2 differential neurons D (numbered 3 and 6), totaling 6 neurons. Their input values come from the output value of the previous input layer multiplied by the corresponding weights, and their output value is the PID algorithm. Its expression is as follows:

$$u_i'(k) = \sum_{i=1}^{3}\sum_{j=1}^{2} w_{ij}x(k) \tag{4}$$

Proportional neurons:

$$x'_{1,4}(k) = u'_{1,4}(k) \tag{5}$$

Differential neurons:

$$x'_{2,5}(k) = x'_{2,5}(k-1) + u'_{2,5}(k) \tag{6}$$

Integral neurons:

$$x'_{3,6}(k) = u'_{3,6}(k) - u'_{3,6}(k) \tag{7}$$

To ensure the stability of the system output, it is necessary to limit the input and output of the PID neuron. According to the normalization operation rules, the upper limit is+1 and the lower limit is -1. The formula is as follows:

$$\lim_{u\to\infty} x'(u) = 1, \lim_{u\to\infty} x'(u) =- 1 \tag{8}$$

3.1.3 Output layer algorithm

The task of the output layer of PIDNN is to complete the total output of the neural network. Its input is the weighted sum of the output values of the previous hidden layer, expressed as:

$$u''_h(k) = \sum_{h=1}^{2}\sum_{j=1}^{3} w_{hj}x'_j(k) \tag{9}$$

Like the hidden layer, for the convenience of normalization calculation, the input and output of the output layer also need to be limited. The expression for outputting $x''_h(k)$ is:

$$x''_h(k) = \begin{cases} 1, u''_h(k) > 1 \\ u''_h(k), u''_i(k) \in [-1,1] \\ -1, u''_h(k) < 1 \end{cases}$$

(10)

In equations (3) to (10), i is the neuron number of the input layer (i=1, 2, 3, 4), j is the neuron number of the hidden layer (j=1, 2, 3, 4, 5, 6), and h is the neuron number of the output layer (h=1,2); w_{ij} is the connection weight value from the input layer to the hidden layer, while w_{hj} is the connection weight value from the hidden layer to the output layer; Variables marked with single quotes represent hidden layer variables, while variables marked with double quotes represent output layer variables.

3.2 Feedback Path Algorithm

The task of neural network reverse path is to continuously modify the weight values of each connection according to the optimal gradient descent method, based on the difference between the system output value and the target value, as well as the values of various state variables, in order to achieve the goal of training the neural network. In this system, we continuously optimize the w_{ij} and w_{hj} values to make the system output closest to the target value.

The goal of PIDNN learning is to find the least squares of the error, even if J reaches the minimum:

$$J = \frac{1}{L}\sum_{i=1}^{m} \|r(k) - y(k)\| = \frac{1}{L}\sum \|e(k)\| \tag{11}$$

In the formula, m is the number of sampling points.

Weight adjustment algorithm for PIDNN neural network:

Hidden layer to output layer:

$$w(n+1) = w_{jh}(n) - \eta'_{jh}\frac{\partial J}{w_{jh}} \tag{12}$$

$$\frac{\partial J}{\partial w_{jh}} =- \frac{2}{m}\sum_{k=1}^{m} [r_h(k) - y_h(k)]\, x'_h(k) \tag{13}$$

Input layer to hidden layer:

$$w_{ij}(n+1) = w_{ij}(n) - \eta_{ij}\frac{\partial J}{\partial w_{ij}} \tag{14}$$

$$\frac{\partial J}{\partial w_{ij}} =- \frac{2}{m}\sum_{k=1}^{m} [r_j(k) - y_i(k)]\, w'_{ij} \times sgn\frac{y_{j+1}(k+1) - y_j(k)}{v_j(k) - v_{hj}(k-1)} \times sgn\frac{x'_{ij}(k) - x'_{ij}(k-1)}{u'_{ij}(k) - u'_{ij}(k-1)} \times x_i(k) \tag{15}$$

4. Experimentation

Taking the car starter QDJ2532B as the test object, according to the above calculation formula and the physical parameters of the starter in Table 1[10][11], the initial weight values of PIDNN are shown in Table 2. Based on this, PIDNN is trained with a learning step size of 0.05 and a sampling number of m=800 for each step. After 15 steps, the training is completed.

Table 1. Parameters Of Starter

parameter	value
resistance (r)/ Ω	0.732
Armature inductance (L)/H	0.212
Electric potential coefficient (Ce)/V·s/rad	0.45
Moment of inertia (J)/J·s2	0.023
Torque coefficient (CT)/N·m/A	0.51

Table 2. Original weight of PIDNN

w_{ij}(Input layer to hidden layer)		
w11=-0.5	w12=-0.1	w13=-4
w14=-0.5	w15=-0.1	w16=-4
w21=-0.45	w22=-0.15	w23=-3.5
w24=-0.45	w25=-0.15	w26=-3.5
w31=-0.5	w32=-0.1	w33=-4
w34=-0.5	w35=-0.1	w36=-4

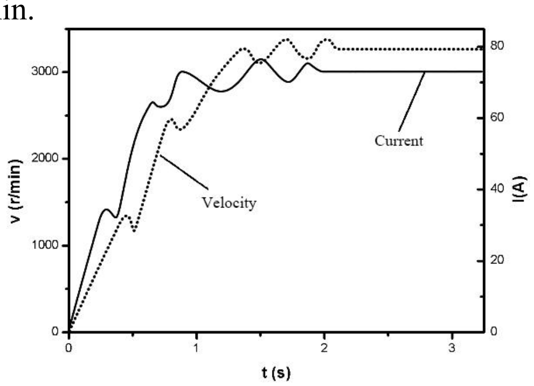
w41=-0.45	w42=-0.15	w43=-3.5
w44=-0.45	w45=-0.15	w46=--3.5
w_{hj}(Hidden layer to output layer)		
w11=1	w12=-1	w13=0.6
w14=-0.6	w15=1	w16=-1
w21=1	w22=-1	w23=0.6
w24=-0.6	w25=1	w26=-1

The core of the control system is an x86 architecture industrial computer with PC104 specifications, which extends Advantech's multifunctional high-speed data acquisition card PCI1716H through PCI bus and uses PLC for switch quantity control. The operating system platform adopts Windows 10, the main control software is VC++2020, and the data sampling cycle is 1ms.

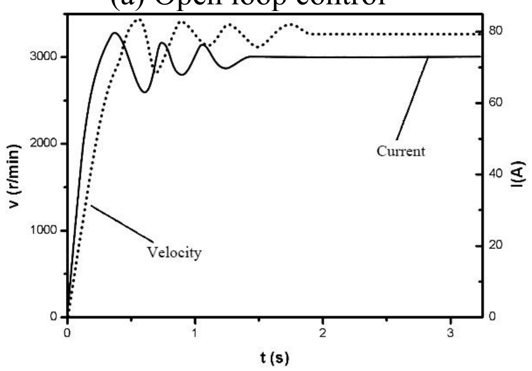
Performance tests were conducted on the system under three different control methods: open-loop, PIDNN with initial weights based on experience, and trained PIDNN.

As shown in Figure 3, the setting adjustment targets for (a), (b), and (c) are a speed of 3200r/min and a current of 72A, respectively, using the output waveforms corresponding to the three controllers mentioned above.

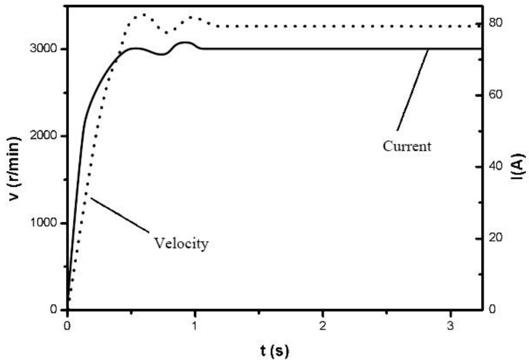
From Figure 3, it can be seen that the current and speed of the open-loop regulation reach steady state at 2.2 seconds, with a maximum overshoot of 80A and 3587r/min; The PIDNN regulator with given initial weights reaches steady state in 1.8 seconds, with a maximum overshoot of 83A and 3910r/min; The trained PIDNN regulator reaches steady state in 1.2 seconds, with a maximum overshoot of 75A and 3850r/min.



(a) Open loop control



(b) PIDNN control with given initial weights



(c) Optimized PIDNN control

Fig. 3 Velocity and current response of start test

The above experiments indicate that the trained PIDNN controller has advantages such as smaller overshoot, shorter transition time, and faster adjustment speed, which can better ensure that the testing process can be completed in a shorter time, better protect the starter, improve testing efficiency and accuracy.

Applying the PIDNN controller to the automotive starter testing system effectively combines the advantages of simple and efficient PID algorithm with the strong adaptive ability of neural networks. The experimental results show that the PIDNN controller exhibits good control advantages, such as fast dynamic response and wide speed regulation range, providing a simple and practical new method for practical engineering applications.

5. Summary

Applying the PIDNN controller to the automotive starter testing system effectively combines the advantages of simple and efficient PID algorithm with the strong adaptive ability of neural networks. The experimental results show that the PIDNN controller exhibits good control advantages, such as fast dynamic response and wide speed regulation range, providing a simple and practical new method for practical engineering applications.

Acknowledgment

We thank all reviews for their constructive comments. This work was supported by a grant from the 2023 Hunan Province Vocational College Education and Teaching Reform Research Project "Research on Community Education Layered Teaching Model Based on Smart Education System". China. 2022 Hunan Provincial Department of Education Scientific Research Project "Research on Visual Detection Enhancement Algorithm for Surface Defects of Permanent Magnet Motor Magnetic Tiles". China (Project No. 22C0972). Key funded project of Hunan Open University in 2023. China (Project No. XDK-2023-A-3).

References

[1] Khaled T A, Akhrif O, Bonev I A. Dynamic path correction of an industrial robot using a distance sensor and an ADRC controller[J]. IEEE/ASME Transactions on Mechatronics, 2020, 26(3): 1646-1656.

[2] Jin H, Song J, Lan W, et al. On the characteristics of ADRC: A PID interpretation[J]. Science China Information Sciences, 2020, 63: 1-3.

[3] Yang Z, Jia J, Sun X, et al. An enhanced linear ADRC strategy for a bearingless induction motor[J]. IEEE Transactions on Transportation Electrification, 2021, 8(1): 1255-1266.

[4] Tian M, Wang B, Yu Y, et al. Discrete-time repetitive control-based ADRC for current loop disturbances suppression of PMSM drives[J]. IEEE Transactions on Industrial Informatics, 2021, 18(5): 3138-3149.

[5] Wu X, Jiang D, Yun J, et al. Attitude stabilization control of autonomous underwater vehicle based on decoupling algorithm and PSO-ADRC[J]. Frontiers in Bioengineering and Biotechnology, 2022, 10: 843020.

[6] Yang Z, Wang Z, Yan M. An optimization design of adaptive cruise control system based on MPC and ADRC[C]//Actuators. MDPI, 2021, 10(6): 110.

[7] Wu Z, Gao Z, Li D, et al. On transitioning from PID to ADRC in thermal power plants[J]. Control Theory and Technology, 2021, 19: 3-18.

[8] Patelski R, Dutkiewicz P. On the stability of ADRC for manipulators with modelling uncertainties[J]. ISA transactions, 2020, 102: 295-303.

[9] Garrido R, Luna L. Robust ultra-precision motion control of linear ultrasonic motors: A combined ADRC-Luenberger observer approach[J]. Control Engineering Practice, 2021, 111: 104812.

[10] Liu L, Liu Y, Zhou L, et al. Cascade ADRC with neural network-based ESO for hypersonic vehicle[J]. Journal of the Franklin Institute, 2023, 360(12): 9115-9138.

[11] Zhan B, Zhang L, Liu Y, et al. Model predictive and compensated ADRC for permanent magnet synchronous linear motors[J]. ISA transactions, 2023, 136: 605-621.