AN ADAPTIVE CONTROLLER BASED ON NEURAL NETWORKS FOR MOTOR-DRIVE LOAD SIMULATOR

Shen Dongkai   Wang Zhanlin
College of Automation Science and Electrical Engineering
Beijing University of Aeronautics and Astronautics Beijing 100083, P.R.China
ridixklh@public3.bta.net.cn

ABSTRACT

A based Radial-Basis-Function (RBF) networks adaptive control scheme is proposed for Motor-driven load simulator in the paper. The system is highly nonlinear and includes delays in the control loop and extraneous force disturbance from load. These factors have many bad effects on the tracing accuracy and dynamic performance of the simulator. Contrasted with the conventional-BP-networks-based controllers, the proposed algorithm is more efficient without the **local-minimal** problem. This controller is robust, efficient and simple. The structure invariance principle is used to reduce the **extraneous force**, but in fact, this method does not work well due to the modeling error and the speed error produced by the speed sensor. The RBF neural networks can be used in adaptive controller for simulator, and the outputs of the neural networks are used as the parameters of the controllers to compensate for the effects of mentioned factor. In this paper, the key problem discussed is to design a load simulator with high tracing accuracy and wide bandwidth. The simulation results presented in the paper shows that the designed controller provides good control performance and adaptive compensation of the extraneous force.

INTRODUCTION

We must control actuator and alter the magnitude of the thrust and the direction of the thrust for controlling aerocraft. The performance of the aerocraft is correlated with the performance of the control system. Such as: the missile is reliable and exact to hit the target quickly. The plane completed various high-difficult mission is dependent on the fly control system. The pneumatic load of the actuator is concerned with the altitude, the velocity and the angle of flight. The tasks of Motor-driven load simulator is realtime to reappear this load spectrum quickly, and bring to bear torque on the moving actuator by rule and line. There is a pivotal matter to eliminate extraneous force in exact tracking of flight control system load spectrum. The fast progress of electron technique accelerate development of motor-control. PM dc torque-motors are utilized to loading system. This improve system efficiency and simplify loading system. Due to highly nonlinear and delays in the control loop and extraneous force disturbance from load in the Motor-driven load simulator, these factors have many bad effects on the tracing accuracy and dynamic performance of the simulator. To overcome this problem, adaptive and model-based control techniques have presented. Although these methods have the ability to deal with structured uncertainties arising from the imprecision in the system model, unknown payloads, and inaccuracies on the torque constants of the actuators, their performance in the presence of unstructured uncertainties that occur due to unmodeled dynamics cannot be guaranteed. The structure invariance principle is used to reduce the extraneous force, but in fact, this method does not work well due to the modeling error and the speed error produced by the speed sensor. In order to overcome these drawbacks, a based Radial-Basis-Function (*RBF)* networks adaptive control scheme is proposed here. *RBF* neural networks can be used in adaptive controller for simulator, and the outputs of the neural networks are used as the parameters of the controllers to compensate for the effects of mentioned factor. The simulation results presented in the paper shows that the designed controller provides good control performance and adaptive compensation of the extraneous force.

THE LOAD SYSTEM OF THE PM DC TORQUE MOTOR CONTROL

In the proposed work, a novel control strategy of a PM dc torque-motor is proposed incorporating an on-line weights and biases updating feature of the neural networks. The neural networks architecture is based on the inverse dynamic model of the nonlinear torque servo system. To enhance the robustness, which is an important criterion of a high-performance load, a feature of adaptive learning rate is introduced.

Although it is not mandatory to obtain a motor model if the neural networks is used in the torque-motor control system, it may be worth doing so from the analytical perspective, in order to establish the foundation of the neural networks structure.

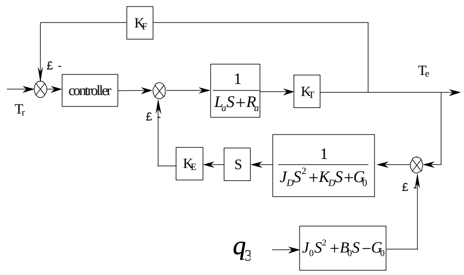

Fig. 1 block diagram of PM dc motor load system

The dynamics of the PM dc torque motor load system can be described by the following equations:

$$U_a(s) = R_a I_a(s) + L_a S I_a(s) + E_a(s)$$

$$T_e(s) = J_D S^2 q_1(s) + K_D S q_1(s) + T_M$$

$$E_a(s) = K_e S q_1(s)$$

$$T_e(s) = K_T I_a(s)$$

$$T_M = G_0(q_1(s) - q_3(s)) + J_0 S^2 q_3(s) + B_0 S q_3(s)$$

where $U_a(s)$, $E_a(s)$, and $I_a(s)$ are the time-varying motor terminal voltage, back EMF, and armature current, respectively. K_e and K_T are the motor back-EMF and torque constants, respectively, R_a and L_a are the armature resistance and inductance, $\dot{q}_1$ and $\dot{q}_3$ are the motor speed and the actuator speed, respectively, T_e, T, and T_M are the developed torque, the extraneous force of load system, and The loading torque, respectively, G_0 is the torsion rigidity, J_D and K_D are the inertia and viscous constants, respectively, and J_0 and B_0 are the inertia and viscous constants of the load, respectively. The relation between the extraneous force and the angle displacement can be described by the following relation:

$$T = G_0(q_1(s) - q_3(s))$$

To make the control task easier, the PM dc motor load system can be express as a single-input single-output system with the disturbance of the positional type by the above equations. The motor parameters and expressions for the above constants are given in the real system. The above equations can be modified to obtain the dynamic model of the load system as follows:

$$T_e(S) = \frac{K_T(J_D S^2 + K_D S + G_0)U_a(S) + (J_0 S^2 + B_0 S - G_0)K_T K_e S q_3(S)}{L_a J_D S^3 + (L_a K_D + R_a J_D)S^2 + (R_a K_D + L_a G_0 + K_T K_e)S + R_a G_0} \quad (1)$$

where the latter is the disturbance of the position type in the numerator (the extraneous force), and derived from the angle displacement $(q_3(S))$of the actuator.

The purpose of using the neural networks is map the nonlinear relationship between the terminal voltage and the output torque of the PM dc torque motor.

THE DESIGN OF BASED NEURAL NETWORKS CONTROLLER

The Radial-Basis-Function networks is used as on-line identification in the paper. Neural networks is used as the dynamic observation to obtain mapping of the output of the plant and the input of the controller. The controller is consist of the *PID* controller and the neural network compensated controller. the latter one compensate the former. It is farther to improve the performance of the load system.

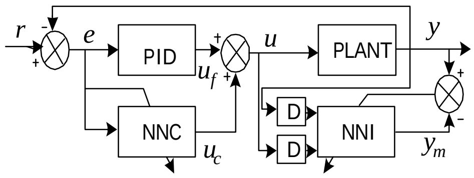

Fig. 2 neural network control system

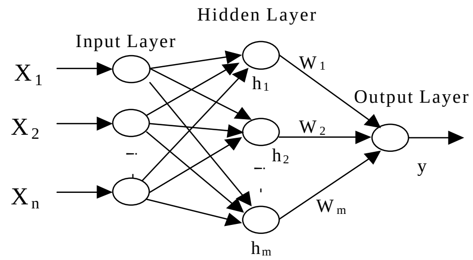

Fig. 3 The *NNI* structure

The architecture of the control system is shown in figure 2, where *r(t)* is a given input, *u(t)* is the output of controller. The *NNC*, the *NNI* are the compensated controller of neural networks, the identification of neural networks, respectively. *D* is the delay tache. The output of the system in the last step and the output of the controller in the last step are input to the *NNI*. $y_m(t)$ is the output of the *NNI.*

The *NNI* structure for torque load system is shown in figure 3. The activation function used in the hidden and output layers are Radial-Basis-Function (*RBF*) and log sigmoid, respectively. Once this basic design of the *NNI* structure is done, the next step is to determine the weights and biases of the *NNI* through the training to achieve the specific target with the given input. The Radial-Basis-vector of the *RBF* neural networks is defined as follows :

$$H = [h_1, h_2, \ldots, h_j, \ldots, h_m]^T$$

The vector h_j can be obtained by using the Guassian type of functions as follows:

$$h_j = \exp\left(-\frac{\|X - c_j\|^2}{2b_j^2}\right),\ j=1, 2, \cdots, m. \quad (2)$$

where $X=[x_1, x_2, \ldots x_n]^T$ is the input vector of the networks. $C_j = [c_{j1}, c_{j2} \ldots c_{ji} \ldots c_{jn}]^T$ is the radial basis function center vectors. $B = [b_1, b_2 \ldots \ldots b_m]^T$ is the shape vector of the radial basis function.

The output of the NNI is defined as follows:

$$y_m(t)=w_1 h_1+w_2 h_2+\ldots \ldots +w_m h_m \qquad (3)$$

where $W=[w_1, w_2 \ldots w_j \ldots w_m]^T$ is the weight vectors.

The identification neural networks (*NNI*) parameter adjust arithmetic is as follows:

The identification performance target function is described by the equation:

$$J_1 = \frac{1}{2}(y(t+1) - y_m(t+1))^2 \qquad (4)$$

The NNI parameters are updated at each instant using the gradient descend algorithm, as follows:

$$\begin{aligned} w_j(t+1) &= w_j(t) + \eta(y(t+1) - y_m(t+1))h_j \\ &+ \alpha(w_j(t) - w_j(t-1)) \end{aligned} \qquad (5)$$

$$\begin{aligned} b_j(t+1) &= b_j(t) + \eta(y(t+1) \\ &- y_m(t+1))w_j h_j \frac{\|X - C_j\|^2}{b_j^3} \\ &+ \alpha(b_j(t) - b_j(t-1)) \end{aligned} \qquad (6)$$

$$\Delta c_{ji} = (y(t+1) - y_m(t+1))w_j \frac{x_j - c_{ji}}{b_j^2} \qquad (7)$$

$$\begin{aligned} c_{ji}(t+1) &= c_{ji}(t) + \eta \Delta c_{ji} \\ &+ \alpha(c_{ji}(t) - c_{ji}(t-1)) \end{aligned} \qquad (8)$$

where η is the learning rate, α is the energy factor. Jacobian matrix (the sensitivity of the input/output of the plant) is described by the following equation:

$$\frac{\partial y(t+1)}{\partial u(t)} = \sum_{j=1}^{j=m} w_j h_j \frac{c_{ji} - x_1}{b_j^2} \qquad (9)$$

where $x_1 = u(t)$.

The Neural Networks Compensated Controller

The structure of the neural networks compensated controller (*NNC*) is similar to one of the *NNI*, but there is the difference, which is that the error signal is converted into the signal of the proportion, the integral, the differentiation as the input signal. It is as follows:

$$xc_1 = e(t)$$

$$xc_2 = e(t) - e(t-1)$$

$$xc_3 = \sum_{k=0}^{k=t} e(k)$$

$$e(t) = r(t) - y(t)$$

The energy function of the weights adjustment is as follows:

$$J_c = \tfrac{1}{2}(r(t+1) - y(t+1))^2 \qquad (10)$$

The output of the *PID* controller is as follows:

$$u_f = xc_1 k_p + xc_2 k_d + xc_3 k_i \qquad (11)$$

where k_p k_i k_d are the coefficient of the proportion, the integral, the differentiation, respectively.

The output of the *NNC* is as follows:

$$u_c = h_{c_1} w_{c_1} + h_{c_2} w_{c_2} + h_{c_j} w_{c_j} + \ldots + h_{c_n} \qquad (12)$$

where ***n*** is the hidden number of the neuron of the NNC, w_{c_j} is the weight.

Therefore, the output of the controller is as follows:

$$u = u_f + u_c \qquad (13)$$

The parameter adjustment algorithm is as follows:

$$\begin{aligned} w_{c_j}(t+1) &= w_{c_j}(t) + \eta_c(r(t+1) \\ &- y(t+1))h_{c_j} \frac{\partial y(t+1)}{\partial u(t)} + \alpha_c \Delta w_{c_j}(t) \end{aligned} \qquad (14)$$

$$\Delta w_{c_j}(t) = w_{c_j}(t) - w_{c_j}(t-1) \qquad (15)$$

$$\begin{aligned} b_{c_j}(t+1) &= b_{c_j}(t) + h_c(r(t+1) - y(t+1)) \\ & w_{c_j} h_{c_j} \frac{\|Xc - C_{c_j}\|^2}{b_{c_j}^3} \frac{\partial y(t+1)}{\partial u(t)} + \alpha_c(b_{c_j}(t) - b_{c_j}(t-1)) \end{aligned} \qquad (16)$$

$$\begin{aligned} \Delta c_{c_{ji}} &= (r(t+1) - y(t+1)) \\ & w_{c_j} \frac{xc_j - c_{c_{ji}}}{b_{c_j}^2} \frac{\partial y(t+1)}{\partial u(t)} \end{aligned} \qquad (17)$$

$$\begin{aligned} c_{c_{ji}}(t+1) &= c_{c_{ji}}(t) + \eta_c \Delta c_{c_{ji}} \\ &+ \alpha_c(c_{c_{ji}}(t) - c_{c_{ji}}(t-1)) \end{aligned} \qquad (18)$$

where η_c is the learning rate. α_c is the energy factor.

In the paper, in order that the performance of the system be accepted, the disturbance of the extraneous force is used as the input of the *NNC* in the electromotion load system.

SIMULATION

In this section simulation results are presented applying the proposed control scheme to control the load system with the position disturbance. The given torque input is as follows:

$$r(t) = \sin(20\pi t)$$

The disturbance of the position type is described as:

$$r(t) = \sin(10pt)$$

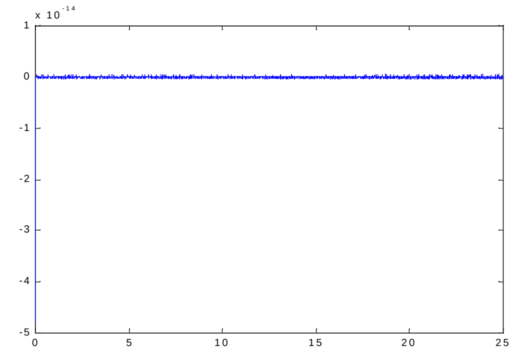

Fig. 4 identification error

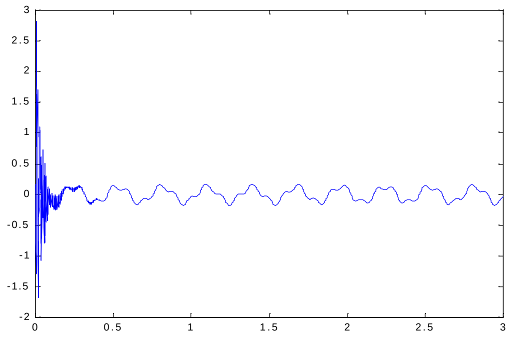

Fig. 5 track error

The 3-6-1 structure is used in the *NNI* structure and the *NNC* structure. The sampling time is 1ms. The identification error of the system is depicted in the Fig. 4.

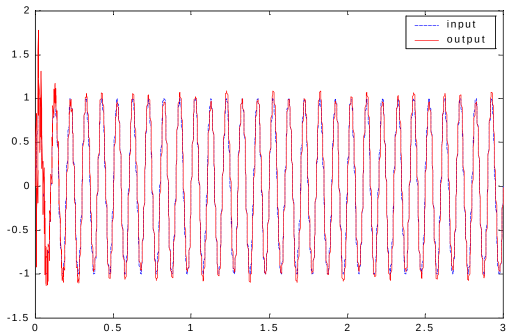

Fig. 6 the torque track with disturbance

In order to validate the effectiveness of the controller, the torque tracking error of the system is illustrated in Fig. 5. The torque tracking performance is shown in Fig. 6. According to the above figure, the tracking error is 9 percent. There is well robustness to parameter variety in the load system. The perfect torque tracking is achieved by implement the proposed control scheme. The elimination of the extraneous force is to 91 percent.

CONCLUSION

In this paper, an adaptive torque controller using RBF neural networks is proposed for the electromotion load system. The control scheme based on RBF networks and an effective on line weight modification algorithm, in which the parameters are directly updated from the measurement of the input, output, and the operating point. It is obtained the better effect at the control. The control strategy overcome the problem of local minima, and the load system reduce the disturbance of extraneous force when RBF network is used in the torque load system. The simulation results have been shown to be very efficient at control and learning.

REFERENCES

1. Fang CongZhi, the system identification, the TsingHua university publishing company, 1987.
2. M. Asano, T. Yamamoto, T. Oki and M. Kaneda A Design of Neural-Net Based Predictive PID Controllers IEEE Trans. On Neural Networks, 1999 pp. 1113-1118
3. Chien-Cheng Lee, Pau-Choo Chung, and Chein-I Chang Robust Radial Basis Function Neural Networks IEEE Trans. ON Systems, Man and Cybernetics—Part B: Cybernetics, Vol. 29, NO. 6 December 1999
4. M. S Ahmed, Neural-Net-Based Direct Adaptive Control For a Class of Nonlinear Plants, IEEE Transactions on automatic control, VOL. 45, NO, 1, January 2000
5. Prasanna Boyagoda, Mutsuo Nakaoka, A servo tracking controller based on neural networks, IEEE 1999 International Conference on Power Electronics and Drive Systems, PEDS'99, July 1999, Hong Kong.
6. Miguel Strefezza, Yasuhiko Dote, Radial Basis Neural Network Adaptive Controller for Servomotor, IEEE, pp. 747-752, 9,1993.
7. Chen Xiaohong, Gao Feng, Qian Jixin, Adaptive Control Based on RBF Networks, IEEE Proceedings of the 35th Conference on Decision and Control, Kobe, Japan December 1996.