International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 04 – Issue 07 || July 2019 || PP. 60-64

Research on Multi-modal Control Strategy of Stepping Motor Position Servo System Based on RBF Neural Network

Fengxia Tian
Jiangmen Polytechnic,
Jiangmen 529090, China

Abstract: The position servo system of stepper motor designed in this paper is based on current control scheme. Its main purpose is to solve the problems of out-of-step and oscillation in open-loop control of stepper motor, so that the position control performance of stepper motor is better. Traditional fuzzy control can improve the fast response performance of the system, but its accuracy is not high. In order to reduce the steady-state error of stepper motor position servo system and improve the response speed of the system, so that the system has both rapidity and accuracy, a multi-mode control strategy based on RBF neural network control is proposed in this paper, that is, a control strategy that can switch automatically according to the system state in real time.
Keywords: Multi-modal Control Strategy, Stepping Motor, Position Servo System, RBF Neural Network.

1. Introduction

The traditional control strategy has been unable to meet the requirements of stepper motor control accuracy because of the higher and higher requirements of various industries. With the rapid development of computer technology and electronic semiconductor technology in recent years, W and various control ideas are constantly put forward and improved, and the control strategy of stepping motor is constantly innovated, and has become a hot research topic in the field of stepping motor. There are two control modes of stepping motor, open-loop control and closed-loop control. The difference between the two control methods is whether there is position feedback or speed feedback. In open-loop control mode, because there is no position feedback, in order to achieve accurate position or speed control, the stepper motor must respond to each excitation change. If the excitation speed is too fast, the motor rotor will not be able to respond to the excitation in time. At this moment, the step motor will be out of step, resulting in position error. Because there is no position feedback, this error will be retained and cannot be corrected. Therefore, it is particularly important to adopt closed-loop control strategy for the step motor.

2. RBF Neural Network

RBF neural network was proposed by J. Moody and C. Darken in the late 1980s firstly. Theoretically, RBF neural network can approximate any continuous function defined on a compact set to any prescribed degree of accuracy by expanding the networks structure sufficiently.

Neural network is known for its strong capacities of self-learning, self-adapting and self-organization, it is suitable for the control of nonlinear systems. Radial basis function (RBF) neural network constitute a special network architecture that presents remarkable advantages over other type neural network including better approximation ability, simpler network structures and faster learning speed. RBF neural networks is continuously increasing its popularity in many fields such as pattern recognition, optimization and control.

Basic RBF neural network consists of three layers, namely input layer, hidden layer, and output layer. The nodes within each layer are fully connected to previous layer. The input nodes are connected to the hidden layer neurons directly. The mapping from input layer to output layer is not linear, but the mapping from hidden layer to output layer is linear, thus accelerate the speed of learning greatly and avoid the local minimum problem. The RBF neural network structure is multi-input and single-output (MISO), RBF neural network structure is represented in Fig.1.

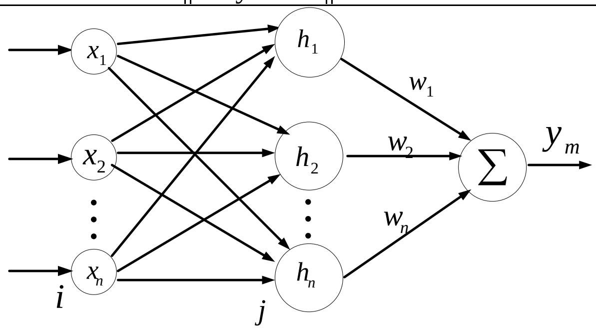

Fig.1 RBF Neural Network structure

In RBF neural network, the input vector is $\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$, assuming radial basis vector is $\boldsymbol{h}=[h_1,h_2,\cdots,h_m]^{\mathrm{T}}$, h_j is Gaussian function,

$$h_j=\exp(-\frac{\| x-c_j^2 \|}{2b_j^2}),\ j=1,2,\cdots m \tag{1}$$

Where c_j is the neuron center and b_j is the center spread parameter. Here, we have used Gaussian transfer function for the hidden neurons. The weight from input layer to hidden layer is 1, the weight vector from hidden layer to output layer is $\boldsymbol{W}=[w_1,w_2,\cdots,w_m]^{\mathrm{T}}$. For the neuron the output is

$$y_m=\sum_{j=1}^{m} w_j h_j \tag{2}$$

The performance index function of identifier is

$$J=\frac{1}{2}(y(k)-y_m(k))^2 \tag{3}$$

According to the gradient descent method, node center, output weights and node base width parameters are shown as the following formula

$$\left.\begin{aligned}
&\Delta w_j(k)=\eta(y(k)-y_m(k))h_j\\
&w_j(k)=w_j(k-1)+\Delta w_j(k)+\alpha(w_j(k-1)-w_j(k-2))\\
&\Delta b_j(k)=\eta(y(k)-y_m(k))w_j h_j\frac{\left\|X-c_j\right\|^2}{b_j^3}\\
&b_j(k)=b_j(k-1)+\Delta b_j(k)+\alpha(b_j(k-1)-b_j(k-2))\\
&\Delta c_j(k)=\eta(y(k)-y_m(k))w_j\frac{x_j-c_{ji}}{b_j^2}\\
&c_{ji}(k)=c_{ji}(k-1)+\Delta\eta c_{ji}(k)+\alpha(c_{ji}(k-1)-c_{ji}(k-2))
\end{aligned}\right\} \tag{4}$$

Where η is the learning rate, α is momentum factor.
Jacobian matrix algorithm is

$$\frac{\partial y(k)}{\partial\Delta u(k)}\approx\frac{\partial y_m(k)}{\partial\Delta u(k)}=\sum_{j=1}^{m} w_j h_j\frac{c_{ji}-\Delta u(k)}{b_j^2} \tag{5}$$

For the parameters of the controlled object are unknown, the control law $\Delta u(k)$ can not be obtained, so directly using the RBF network in the form, we get the RBF network generalized predictive controller as following

$$\Delta u(k) = \boldsymbol{S}(\boldsymbol{Z}_u(k))\boldsymbol{\theta}_u(k) \tag{6}$$

To ensure the output of the controlled object tracking the given, let the tracking error is:

$$e(k) = y(k) - y_r(k) \tag{7}$$

The parameter $\boldsymbol{\theta}_u(k)$ in the control law $\Delta u(k)$ is adjusted adaptively based on the error $e(k)$, let the identification algorithm of the adaptive law $\boldsymbol{\theta}_u(k)$ is

$$\boldsymbol{\theta}_u(k) = \begin{cases} \hat{\varphi}(k) & |\hat{\varphi}(k)| \le M_u \\ P\{\hat{\varphi}(k)\} & |\hat{\varphi}(k)| > M_u \end{cases} \tag{8}$$

$$\hat{\varphi}(k) = \boldsymbol{\theta}_u(k-N) - \gamma \frac{\xi(Z_u(k-N))}{1+\xi^T(Z_u(k-N))\xi(Z_u(k-N))} e(k) \tag{9}$$

Where γ is the self-adaptive learning rate that is decided by designers. The projection operator $P\{*\}$ is defined.

$$P\{\hat{\varphi}(k)\} = M_u \frac{\hat{\varphi}(k)}{|\hat{\varphi}(k)|} \tag{10}$$

M_u in the equation is decided by designers.
Summing up the above, RBF network direct generalized predictive control strategy is as following
STEP 1.Select the initialization parameters N, N_u and $\boldsymbol{\theta}_u(k-N)$;
STEP 2.Calculate $e(k)$;
STEP 3.Calculate $\boldsymbol{\theta}_u(k)$;
STEP 4.Calculate $\Delta u(k)$, and then work out the control law $u(k)$;
STEP 5. $k = k+1$, return to Step 2.

3. Simulation Analysis

Then the load suddenly changed at constant speed. The motor speed maintain constant at this condition and the load vary. Initially, $2\ \mathrm{N \cdot m}$ load is applied to the motor. Then the load change from $2\ \mathrm{N \cdot m}$ to $5\ \mathrm{N \cdot m}$. The speed response graphs is shown in Fig.2 and Fig.3. The results are summarized. If conventional controller is used, when the load is changed, the speed decrease 3%, recovery time is about 0.2 s. If the RBF neural network controller is used, if the load is changed, the speed decrease 1%, recovery time is 0.05 s.

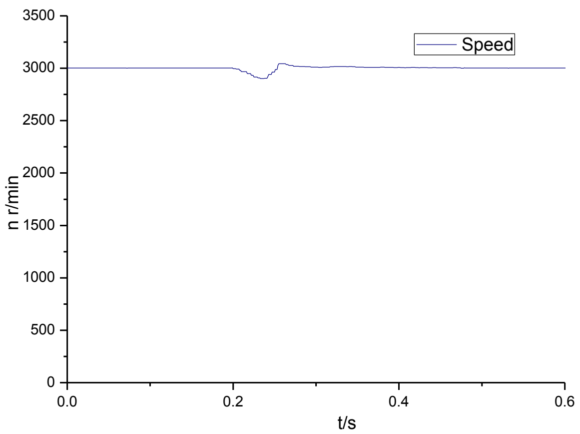

Fig.2 Speed response with load change with conventional method

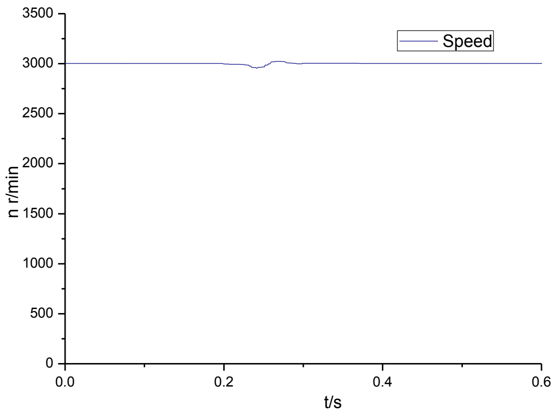

Fig.3 Speed response with load change with RBF neural network

Consider the following nonlinear system

$$y(k)=2.5y(k-1)y(k-2)/[1+y^2(k-1)+$$
$$y^2(k-2)]+1.2u(k-1)+1.4u(k-2)+0.7$$
$$\sin(0.5(y(k-1)+y(k-2)))\cos(0.5(y(k-1)$$
$$+y(k-2)))$$

Reference sequence $y_r(k)$ takes the form of $2+\sin(k\pi T/20)$, $T=0.01$, $N=2$, $N_u=1$, $W_1=50$; adaptive ratio $\gamma=0.001$, each component of the initial values of parameter vector $\boldsymbol{\theta}_u(-1)$ and $\boldsymbol{\theta}_u(0)$ all takes zero; $M_u=3.6$. RBF neural network system selects a routine RBF network, the input of it is

$$\boldsymbol{Z}_u(k)=[\tilde{y}_r(k+2)\quad \tilde{y}_r(k+1)\quad y(k)\quad y(k-1)\quad y(k-2)\quad \Delta u(k-1)]^T,$$

First normalizing the elements in the Input $\boldsymbol{Z}_u(k)$ $\bar{x}=\dfrac{x}{|x|+1}$

Set the number of hidden nodes be 5, the center and width of Gaussian basis function take $\boldsymbol{\mu}=[1\quad 0.6\quad 0.2\quad -0.2\quad -0.6\quad -1]$, $\sigma_i=1.5$.

Simulation result is shown in Fig.4 and Fig.5.

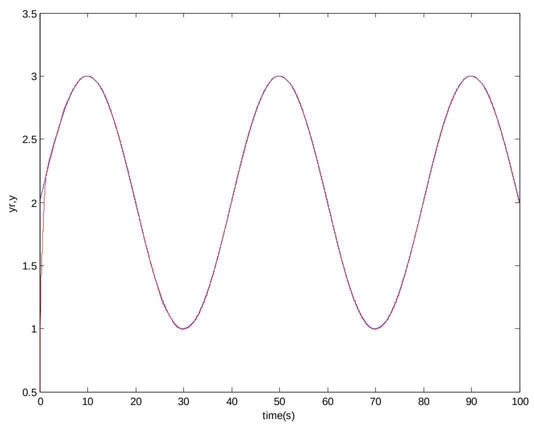

Fig.4 The curve of tracks

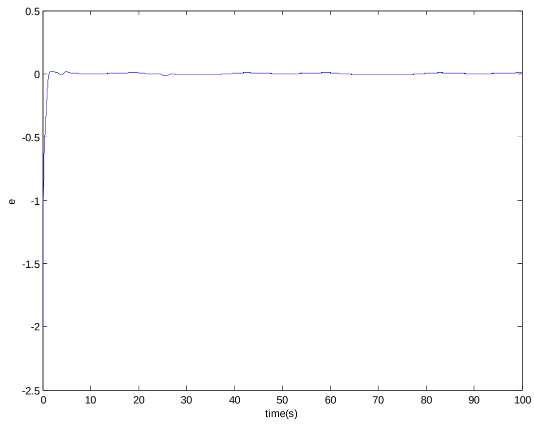

Fig.5 The curve of $e(k)$

4. Conclusion

Starting from the existing stepping motor control strategy, an intelligent control strategy is adopted for the stepping motor system. A multi-mode control strategy based on RBF neural network is proposed, which optimizes the response speed and control accuracy of the stepping motor position servo system and improves the overall position response performance of the system.

References

[1] Liu H, Ouyang P, Wang S. Fault Detection Based on RBF Neural Network in a Hydraulic Position Servo System// World Congress on Intelligent Control & Automation. 2006.

[2] Yuan R, Sun C, Ba S, et al. Analysis of Position Servo System of Pneumatic Manipulator Based on RBF Neural Network PID Control// International Conference on Web Information Systems & Mining. 2011.

[3] Li D H, Chen Z X, Zhai S. Double Closed-Loop Controller Design of Brushless DC Torque Motor Based on RBF Neural Network. Advanced Materials Research, 2012, 503-504:1351-1356.

[4] Liu H M, Wang S P, Ouyang P C. Fault Diagnosis in a Hydraulic Position Servo System Using RBF Neural Network. Chinese Journal of Aeronautics, 2006, 19(4):346-353.

[5] Chen H, Wo S. RBF neural network of sliding mode control for time-varying 2-DOF parallel manipulator system. Mathematical Problems in Engineering,2013,(2013-12-2), 2013, 2013(4):1-10.

[6] Wang X. Modeling of hysteresis for fast tool servo system based on RBF neural network. Dongnan Daxue Xuebao, 2012, 42:217-220.