Scientific Research

The Design of Fractional Order PI$^{\alpha}$D$^{\beta}$Controller Based on Neural Network

Wen Li, Bing Nie, Ming Huang
Software Institute, Dalian Jiaotong University, Dalian, China
Email: lw6017@vip.sina.com

Abstract: A fractional order controller based neural network is proposed, the design method and control algorithm of the controller are discussed. Because introducing neural network, parameters or coefficients of the controller can be adjusted automatically. Control experiments also show that the construct method of the controller and learning algorithm are available to control more complex system.

Keywords: fractional order controller; neural network; control algorithm

1. Introduction (Heading 1)

Fractional calculus is the term given to mathematics involving integral and differential terms of non-integer order. Because the real objects are generally fractional, the fractional order controller can supply a better describing for control system. With combining other intelligent control approaches, the fractional order control would make system have more suitable performance[1-5]. In this paper a fractional order controller based neural network is designed, parameters of the controller can be adjusted automatically. The simplified diagram of experiment system is given in Fig.1, it can be seen as a three-mass-model. Table 1 shows the parameters in Fig.1.

Table 1. System Parameters

1	*Kg*	the elastic coefficient of gear
2	*Ks*	the elastic coefficient of shaft
3	δ	the gear backlash

2. Some ideas about design

Many researches about characters and implementations of fractional order controller are reported, but researches of adaptable fractional order were not much. In practical systems circumstance uncertainty, imprecise system model, etc. are contained usually, so the research on adjusting fractional order parameter real-time is meaningful. In addition PID controllers are still adopted in practice applications, so the fractional order PID controller based on neural network is discussed in this paper.

The PID fractional order controller based neural network is a hybrid controller integrating normal PID, fractional order and neural network control mechanisms. Their combinations can be given with following several formals:

1) The parameters of PI$^{\alpha}$D$^{\beta}$ controller, K_p , K_i , K_d are modifying by means of neural network;

2) The parameters of PI$^{\alpha}$D$^{\beta}$ controller, α, β are modifying by means of neural network;

3) The all parameters of PI$^{\alpha}$D$^{\beta}$ controller are modifying by means of neural network.

There are parameter modifying mode, that is, real-time adjusting and non-real-time adjusting.

In next section the implementation method of PI$^{\alpha}$D$^{\beta}$ controller is proposed, in which all parameters can be modified.

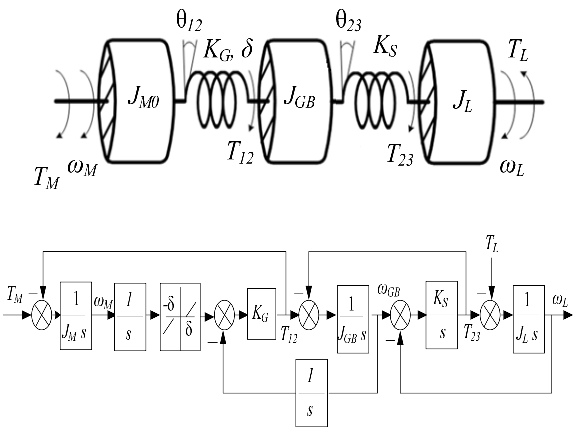

Fig. 1 The simplified diagram of three-mass model

3. The design of fractional order controller based on neural network

3.1 Controller design

The transform function of fractional order PI$^{\alpha}$D$^{\beta}$ controller can be described with Eq.(1):

$$G(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s^{\alpha}} + K_d s^{\beta} = \frac{K_p s^{\alpha} + K_i + K_d s^{\beta+\alpha}}{s^{\alpha}}. \quad (1)$$

From Eq. (1), there is equation as below:

This paper is supported by National Natural Science Foundation of P. R. China (60870009)

978-1-935068-40-2 © 2010 SciRes.

$$u^{\alpha}(t)=K_p e^{\alpha}(t)+K_i e(t)+K_d e^{\lambda}(t) \qquad (2)$$

where 0<α, β<=1, λ=α+β.
According to the direct discretization method of Grűnwald-Letnikov fractional-order differentiation[6-7],

$$ {}_aD_t^{\chi} y(t)\approx \frac{1}{h^{\chi}}\sum_{j=0}^{n} w_j^{\chi} y(t-jh)$$

$$= {}_aD_t^{\chi} y(t)\approx \frac{1}{h^{\chi}}\left[y(t)+\sum_{j=1}^{n} w_j^{\chi} y(t-jh)\right] \qquad (3)$$

where $w_0^{\chi}=1$, $w_j^{\chi}=\left(1-\frac{\chi+1}{j}\right)w_{j-1}^{\chi}$, h is sampling priod, $n=\left[\frac{t-a}{h}\right]$, the notation [·]denotes getting the integer part. From Eq.(3), the $u^{\alpha}(t)$, $e^{\alpha}(t)$ and $e^{\lambda}(t)$ of Eq.(2) can be presented as:

$$u^{\alpha}(t)\approx\frac{1}{h^{\alpha}}u(t)+\frac{1}{h^{\alpha}}\sum_{j=1}^{n} w_j^{\alpha}u(t-jh), \qquad (4)$$

$$e^{\alpha}(t)\approx\frac{1}{h^{\alpha}}\left[e(t)+\sum_{j=1}^{n} w_j^{\alpha}e(t-jh)\right], \qquad (5)$$

$$e^{\lambda}(t)\approx\frac{1}{h^{\lambda}}\left[e(t)+\sum_{j=1}^{n} w_j^{\lambda}e(t-jh)\right]. \qquad (6)$$

Substituting Eq.(4)-Eq.(6) into Eq.(2), the numerical solve of Eq.(2) can be given as below:

$$u(t)=K_p\left[e(t)+\sum_{j=1}^{n} w_j^{\alpha}e(t-jh)\right]+h^{\alpha}K_i e(t)$$

$$+\frac{h^{\alpha}}{h^{\lambda}}K_d\left[e(t)+\sum_{j=1}^{n} w_j^{\lambda}e(t-jh)\right]+\sum_{j=1}^{n} w_j^{\alpha}u(t-jh)$$

$$=K_pA(\alpha)+K_iB(\alpha)+K_dC(\lambda)+D(\alpha). \qquad (7)$$

From Eq.(7), the fractional order $PI^{\alpha}D^{\beta}$ controller based on neural network can be structured and is shown as Fig.2. The system block diagram is given in Fig. 3. For the controller following performance index function is used:

$$E=\frac{1}{2}\sum_{j=1}^{n}[u(t)-u^{*}(t)]^2 . \qquad (8)$$

The modifying algorithm of controller's parameters K_p, K_i, K_d, α and λ can be derived from Eq. (8) with traditional gradient descendent. According to formula $w_j^{\chi}=\left(1-\frac{\chi+1}{j}\right)w_{j-1}^{\chi}$, it is seen that for the weights between input layer and hidden layer, only two weights, $w_1(\alpha), w_1(\lambda)$, need to be modified, because others can be derived from the two weights.

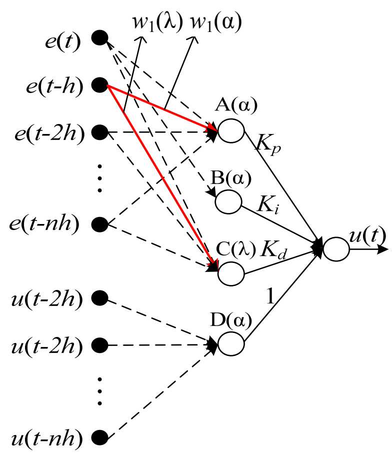

Fig.2 FOC based NN

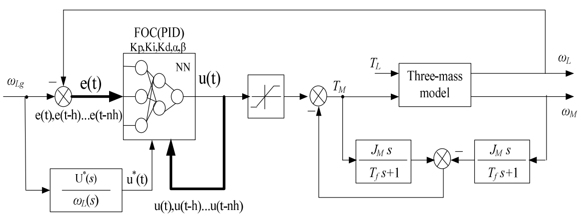

Fig.3 System block diagram

3.2 The Implement of PI^{α} Controller

According above discussing, a fractional order PI^{α} based on neural network is designed for control experiments of the three-mass system. The algorithm of the controller is introduced as follows:
The controller is constructed by a 4-5-3 neural network, the input of input layer is

$$O_j^{(1)}=x(j), \qquad j=1,2,3,4 . \qquad (9)$$

$$X=(x_1,x_2,x_3,x_4)=(e,e_{-1},ce,ce_{-1}) \quad , \qquad (10)$$

the input and output of hide layer are expressed as

$$net_i^{(2)}(k)=\sum_{j=1}^{4} w_{ij}^{(2)}O_j^{(1)}, \quad , \qquad (11)$$

$$O_i^{(2)}(k)=f(net_i^{(2)}(k)), \qquad i=1,2,3,4,5 , \qquad (12)$$

the Sigmoid function is selected as the active function in hide layer, the expression is

$$f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}, \qquad i=1,2,3,4,5 , \qquad (13)$$

the input and output of output layer are

$$net_l^{(3)}(k)=\sum_{i=1}^{5} w_{li}^{(3)}O_i^{(2)}, \qquad (14)$$

$$O_i^{(3)}(k)=g(net_l^{(3)}(k)), \quad l=1,2,3 \qquad (15)$$

$$O=(O_1,O_2,O_3)=(K_p,K_i,\alpha) . \qquad (16)$$

the three outputs of neural network are the three changeable parameters K_p, K_i and α. Because they can not be negative and $0 < \alpha \le 1$, the Sigmoid non-negative function

$$g(x) = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}} \tag{17}$$

is chosen as the active function $g(net_l^{(3)}(k)),\ l = 1,2,3$. The function Equ. (18) is selected as the index function

$$E(k) = \frac{1}{2}(rin(k) - yout(k))^2 \ , \tag{18}$$

the weights are modified by grade descend method, that is,

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \lambda \Delta w_{li}^{(3)}(k-1) \ , \tag{19}$$

where, η is the learning ratio, λ is the inertia coefficient and term $\frac{\partial E(k)}{\partial w_{li}^{(3)}}$ is equal to

$$\frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial net_l^{(3)}(k)} \cdot \frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}(k)} . \tag{20}$$

By analysis and formula derivation, the learning algorithm of output layer can be written as below,

$$\Delta w_{li}^{(3)}(k) = \lambda \Delta w_{li}^{(3)}(k-1) + \eta \delta_l^{(3)} O_i^{(2)}(k) , \tag{21}$$

where $\delta_l^{(3)}$ can be calculated by

$$e(k)\,\mathrm{sgn}\left(\frac{\partial y(k)}{\partial u(k)}\right)\left(\frac{\partial u(k)}{\partial O_l^{(3)}(k)}\right) g'\left(net_l^{(3)}(k)\right),\ \ l = 1,2,3 \ . \tag{22}$$

Similarly the learning algorithm of hide layer can be given,

$$\Delta w_{ij}^{(2)}(k) = \lambda \Delta w_{ij}^{(2)}(k-1) + \eta \delta_i^{(2)} O_j^{(1)}(k) , \tag{23}$$

where $\delta_i^{(2)}$ can be calculated by

$$\left[1 - f^2\left(net_i^{(2)}(k)\right)\right] \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k),\ \ i = 1,2,3,4,5 \ . \tag{24}$$

usually the inertia coefficient λ is selected around 0.9 and the learning ratio is selected between 0.01 and 1. The initial weights are selected at the level of $\frac{1}{\sqrt{n_w}}$, n_w is the number of nodes linked in the former layer[8].

4. Control Experiments

In order to verify the fractional order PI$^{\alpha}$ controller based on neural network, control experiments were conducted with the controller designed on the experimental system given in Fig.4, some parameters of the system are listed in Table.1. . The controller was realized by C language on RT Linux OS. Two groups of experiments were designed to test the parameter tuning performance of the controller. One group is without additional friction and the other is with the friction.

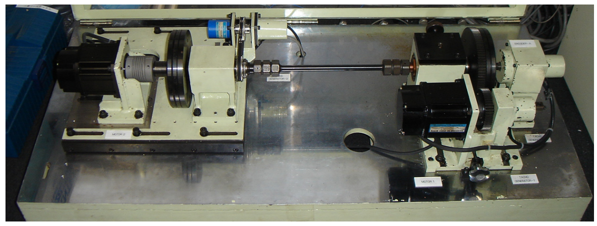

Fig.4. Experimental system

Table 1. System parameters

K_G	3000 (Nm / rad)
K_S	1.9849×10^2 (Nm / rad)
J_{GB}	3.3622×10^3 (Kg·m^2)
J_L	2.9205×10^3 (Kg·m^2)
J_M	4.0156×10^3 (Kg·m^2)

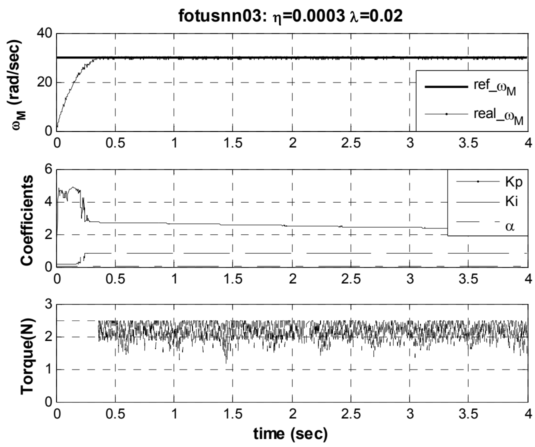

Fig. 5 Control experiment curves without friction

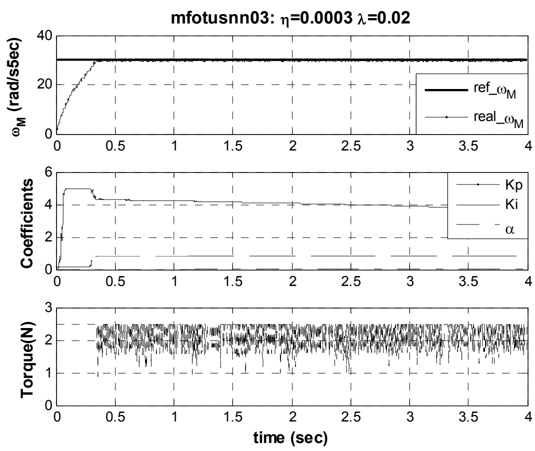

Fig.6 Control experiment curves with friction

Some variables and parameters were recorded on line, for example, setting and real angular velocity ref_ ω_M and real_ ω_M , output torque T_M , proportion and

978-1-935068-40-2 © 2010 SciRes.

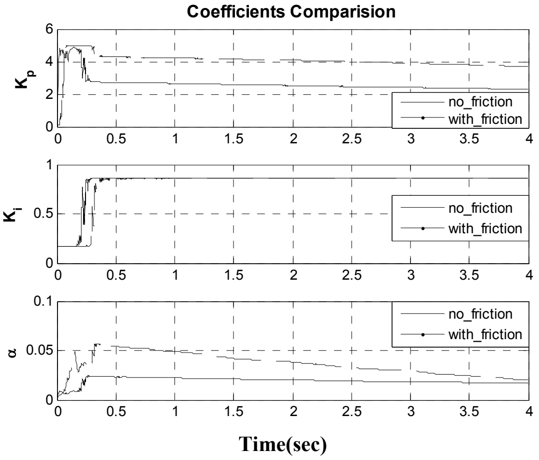

Fig. 7 Kp, Ki, α comparision

integration coefficients K_p and K_i, and fractional order α. According those sampling data, we can plot following curves by MATLAB programming. In titles of Fig.5 and Fig.6, fotusnn03 and mfotusnn03 are filenames of data without friction and with friction respectively. Fig.5 and Fig.6 show that the output torque of system with additional friction changes between 1 to 2.5 (N) to keep real_ ω_M steady around ref_ ω_M, the change of output torque is bigger than that without friction. Fig.7 gives the changing curves of coefficients K_p, K_i and α with different friction. Parameters K_p and α can be increased with friction adding automatically by means of neural network.

5. Conclusion

From above analysis and discussion, it's quite obvious that the fractional order PI^{α} controller based neural network proposed in this paper has attractive flexibility than traditional PI controller, and the neural network introduced makes parameters or coefficients tuned automatically. Control experiments also show that the construct method of the controller and learning algorithm are available to control more complex system.

References

[1] Ivo Petras, Blas M.Vinagre,"Practical application of digital fractional-order controller to temperature control," Acta Montanistica slovaca, vol. 7, 2002, pp. 271–350.
[2] Y.Q.Chen, B.M.Vinagre, and I.Podlubny, Fractional order disturbance observer for robust vibration suppression, Nonlinear Dynamics, 2004, vol. 38, pp. 355-367.
[3] DY Xue, CN Zhao, "Fractioanl order PID controller design for fractional order system," Control Theory and Applications, 2007, Vol. 24, pp.771-776.
[4] W. Li, Y. Hori, "An algorithm for extracting fuzzy rules based on RBFneural network," IEEE Transactions on Industrial Electraonics, vol. 53, pp. 1269–1276, August 2006.
[5] W. Li, Y. Hori, "Vibration suppression using neuron-based PI fuzzy controller and fractional-order," IEEE Transactions on Industrial Electraonics, vol. 54, pp. 117–126, february 2007.
[6] I.Podlubny, Fractional-order systems and controllers. IEEE Transactions on Automatic Control, 1999, Vol.44, pp.208-214.
[7] I.Podlubny, "Fractional differential equations," New York: academic Press, inc. 1999.
[8] H. Lari-Najafi, M. Nasiruddin, "effect of initial weights on back-propagation and tis variations," IEEE International Conference on Systems, Man and Cybernetics, Conference Proceedings, Vol.1, pp. 218-219, Nov. 1989.