Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 5 (5): 892-896, 2005
ISSN 1812-5654

Dynamical Load Emulation Using Neuro-fuzzy Controllers

[1]Muammer Gökbulut and [2]Beşir Dandil
[1]Department of Electronics and Computer Science,
[2]Department of Electrical Science, Faculty of Technical Education, Firat University, Turkey

Abstract: Dynamometer control provides an aid to testing of speed or position control of electrical drives under nonlinear mechanical loads in laboratory environment. This study proposes neuro-fuzzy control of a dynamometer for the emulation of dynamical loads. Varieties of dynamic load models which are the nonlinear function of the shaft speed or position are successfully emulated. Simulation results show that the excellent dynamometer control performance is obtained using the Neuro-Fuzzy Controllers (NFC).

Key words: Neuro-fuzzy controllers, load emulation, Permanent Magnet Synchronous Motor (PMSM)

INTRODUCTION

The load emulation can be used for the controller design and the testing of electrical machines under the complex load conditions, in laboratory environment[1-4]. The main elements of the emulation are the dynamometer and its control system. Emulation set include an electrical drive and a dynamometer which is mechanically coupled to the electrical drive. Permanent Magnet Synchronous Motors (PMSM) are used for the emulation set. Load emulation involves the torque control of the dynamometer such that, the drive machine will see a load which is equal to a desired load model. The simplest method for the dynamic load emulation is to use the inverse load model to calculate the dynamometer torque. It is noted that simulations of inverse model approach are often successful. However, in practice, noise considerations prohibit the use of small time steps for the computation of the inverse dynamics and discretization effects lead to stability problems. Further, it may not always be possible to derive the inverse dynamics of some nonlinear loads[5]. The other approach is to find the right load machine torque by minimization of the error between the desired load model position and the actual shaft position to overcome the problems mentioned above[6].

In recent years, neural network-based fuzzy systems are used in the control of nonlinear systems including electrical drives[7-10]. Although, these studies show the effectiveness of the neural-fuzzy controllers for the electrical drives, the neuro-fuzzy control strategy is not applied to the emulation of nonlinear mechanical loads. This study presents the neuro-fuzzy control of the dynamometer which is directly coupled to the drive machine in order to emulate nonlinear dynamical loads. It aims to overcome the design complexity of the conventional control methods, using the approximation, learning and generalization capabilities of neuro-fuzzy systems. The emulation is placed in the closed loop speed control system of the drive machine.

NEURO-FUZZY LOAD EMULATION FOR PMSM

A permanent magnet synchronous motor is basically synchronous machine with a permanent magnet in the rotor circuit. The armature windings, which are mounted on the stator, are electronically switched according to position of the rotor. The state space model of a PMSM motor referred to the rotor rotating reference frame is given in Eq. 1.

$$\frac{d}{dt}\begin{bmatrix} i_q \\ i_d \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\omega & -\frac{\lambda}{L} & 0 \\ -\omega & -\frac{R}{L} & 0 & 0 \\ \frac{3P^2\lambda}{8J} & 0 & -\frac{B}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} i_q \\ i_d \\ \omega \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 & 0 \\ 0 & \frac{1}{L} & 0 \\ 0 & 0 & -\frac{P}{2J} \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_q \\ v_d \\ T_L \end{bmatrix} \tag{1}$$

Corresponding Author: Muammer Gökbulut, Department of Electronics and Computer Science, Faculty of Technical Education, Firat University, Turkey
E-mail: mgokbulut@firat.edu.tr bdandil@firat.edu.tr

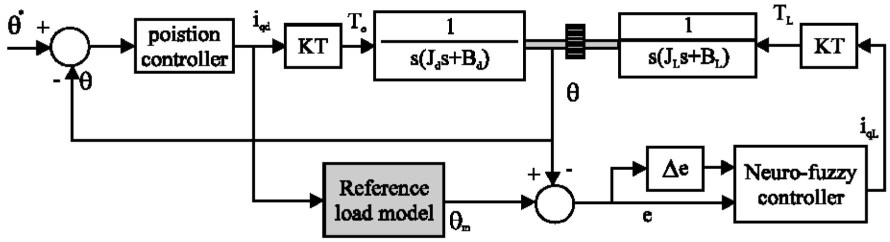

Fig. 1: Neuro-fuzzy load emulation for IM

Where, i_d and i_q is the direct and quadrature components of the stator current, R is the stator resistance, L is the inductance, λ is the magnitude of the flux linkage established by the rotor magnet and ù is the electrical rotor speed, T_L is the load torque (Nm) and J and B are inertia and friction of the motor, respectively. The model is based on the assumptions that the air-gap is uniform, rotor induced current are neglected due to the high resistivity of magnet and stainless steel and the motor is supplied by a three phase sinusoidal input source[11]. Furthermore, PMSM has a surface mounted permanent magnet and thus the d-axis inductance is assumed to be equal to the q-axis inductance. Stator currents are decomposed into the flux and torque components which can be controlled independently for the vector control of electrical drives. The output of the speed controller is the quadratic current component i_q^* (torque current) and direct current component i_d^* (exciting or flux current) is set to zero to avoid demagnetization of the permanent magnet on the rotor.

Thus, two control loops exist which are current and speed control loops. Hence, suitable controllers should be designed for the drive machine speed and current loops and also for the dynamometer current loop since the aim of the load emulation is to control the dynamometer torque to provide a desired load model. For the vector controlled electrical drives, it can be assumed that the current controllers are sufficiently good, i.e., $i_q^*=i_q$ and $i_d^*=i_d$. Then the simplified block diagram for the emulation can be given as in Fig. 1.

Let the inertia $J_m(\cdot)$ and friction $B_m(\cdot)$ of the reference load model be a nonlinear function of the total inertia, friction and the position, respectively. We can write the reference load model as:

$$T_e = J_m(.)\frac{d^2}{dt^2}\theta_m + B_m(.)\frac{d}{dt}\theta_m \qquad (2)$$

where, θ_m is the reference load model position. The purpose of the neuro-fuzzy load emulation is to design a neuro-fuzzy controller for the dynamometer to minimize the performance criteria depending on the position error between the reference load model and motor. Note that the type of the nonlinearity in the reference load model does not matter and the parameters of the drive system do not need to be known when the neuro-fuzzy control is considered for the emulation. Hence, $J_m(.)$ and $B_m(.)$ of the reference load model can be taken as a common static and/or dynamic nonlinear function of the shaft inertia, friction and position, i.e., $J_m(.) = f(J,\theta,\dot{\theta})$ and NFC uses the model following error and the change of error as inputs and produces the torque current for the dynamometer. If the model following error $e(t) = \theta_m(t)-\theta(t)$ is kept close to zero, it actually means that good load emulation is obtained.

NEURO-FUZZY CONTROLLERS

A four layer NFC based on singleton fuzzy inference system, which comprises the input, membership, rule and output layers is adopted to implement the neuro-fuzzy controller[7]. NFC layers perform the fuzzification, inference and defuzzification process of the fuzzy systems, respectively as shown in Fig. 2.

NFC inputs are the position error and the change of the position error such as: $x_1^1 = e(t)$, $x_2^1 = \Delta e(t)$.

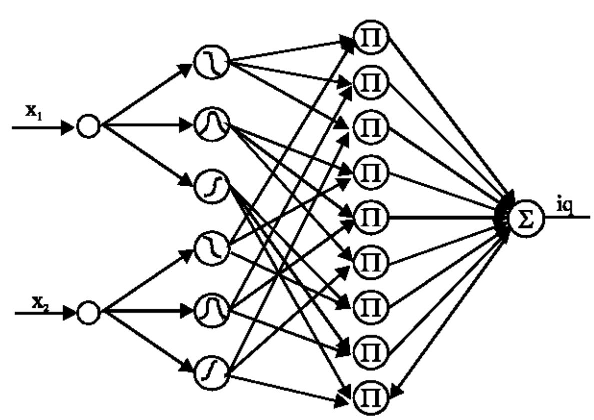

Fig. 2: Neuro-fuzzy controller

For every node i in the input layer, net input and the net output are given as

$$net_i^1 = x_i^1, y_i^1 = f_i^1(net_i^1) = x_i^1 \quad (3)$$

Each node in the membership layer performs a membership function. Bell and sigmoid functions are adopted as membership functions and the net input and output of this layer is,

$$net_j^2 = -s_{ij}\left(x_i^2 - m_{ij}\right),$$

$$y_j^2 = f_j^2(net_j^2) = \frac{1}{1+\exp(net_j^2)} \; j = 1,3 \quad (4)$$

$$net_j^2 = \left|\frac{x_i^2 - m_{ij}}{s_{ij}}\right|^{2b_i},$$

$$y_j^2 = f_j^2(net_j^2) = \frac{1}{1+net_j^2} \; j = 2 \quad (5)$$

where, σ_{ij}, m_{ij} and b_i are the parameters of the membership functions to be determined by training. The third layer of the NFC includes the fuzzy rule base and the each node k in this layer represented by Π determines a fuzzy rule. The output of this layer is given as:

$$net_k^3 = \prod_j w_{jk}^3 x_j^3, y_k^3 = f_k^3(net_k^3) = net_k^3 \quad (6)$$

where, the weights w_{jk}^3 is unity. The single node o in the output layer produces an output as the sum of the inputs. Hence, the output of this layer is:

$$net_o^4 = \sum_k w_{ko}^4 x_k^4, y_o^4 = i_{qL}^* = f_o^4(net_o^4) = net_o^4 \quad (7)$$

where, the connecting weight w_{ko}^4 is the output action strength associated with the kth rule.

Training of the neuro-fuzzy controller: NFC has two groups of parameters to be adapted which are the membership functions and the output parameters as in Eq. 3-7. For the on-line training of the NFC, speed tracking error and the cost function is:

$$e(k) = \theta_m(k) - \theta(k) \quad E(k) = \frac{1}{2}e^2(k) \quad (8)$$

where, θ_m is the load model position, θ is the actual shaft position and e is the position error. According to back propagation learning algorithm, the adaptation law of the weights in the output layer can be defined as:

$$w_{ko}^4(k) = w_{ko}^4(k-1) - \eta\left(\frac{\partial E(k)}{\partial w_{ko}^4}\right) \quad (9)$$

where, the η is the learning rate and $\left(\frac{\partial E(k)}{\partial w_{ko}^4}\right)$ is:

$$\begin{aligned}\frac{\partial E(k)}{\partial w_{ko}^4} &= \frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \theta(k)}\frac{\partial \theta(k)}{\partial i_{qL}(k)}\frac{\partial i_{qL}(k)}{\partial w_{ko}^4} \\ &= -e(k).\frac{\partial \theta(k)}{\partial i_{qL}(k)}.\frac{\partial i_{qL}(k)}{\partial w_{ko}^4}\end{aligned} \quad (10)$$

where, $\frac{\partial i_{qL}}{\partial w_{ko}^4}$ is easily calculated from the Eq. 7. However, $\frac{\partial \theta}{\partial i_{qL}}$ should be calculated using the dynamometer. The gradient of complex nonlinear dynamic systems cannot be found explicitly. Therefore, the approximate value of gradient should be used (i.e., discrete derivative of the system output with respect to the input or sign of the discrete derivative).

$$\frac{\partial \theta(k)}{\partial i_{qL}(k)} = \mathrm{sgn}\left(\frac{\theta(k) - \theta(k-1)}{i_{qL}(k) - i_{qL}(k-1)}\right) \quad (11)$$

In similar way, parameters of the membership functions can be adapted using the gradient descent method such as:

$$\frac{\partial E(k)}{\partial m_{ij}} = \frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \theta(k)}\frac{\partial \theta(k)}{\partial i_{qL}(k)}\frac{\partial i_{qL}(k)}{\partial y_k^3}\frac{\partial y_k^3}{\partial y_j^2}\frac{\partial y_j^2}{\partial m_{ij}} \quad (12)$$

where, the derivatives in Eq. 12 can be calculated using the Eq. 4-7.

SIMULATION RESULTS

The neuro-fuzzy load emulation system given in Fig.1 is simulated for the permanent magnet synchronous motor-dynamometer set which has the nominal parameters $J=1*10^{-4}$ kg m^2, $B=1*10^{-5}$ N ms, λ=0.11 Wb/m^2, R=11.5 ohm, L=0.0215 H and P=6. NFC is trained using pattern

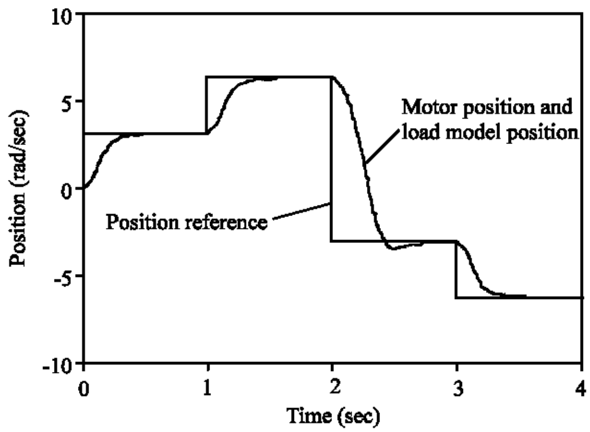

Fig. 3: The model and the shaft position for the load model given by Eq. 13

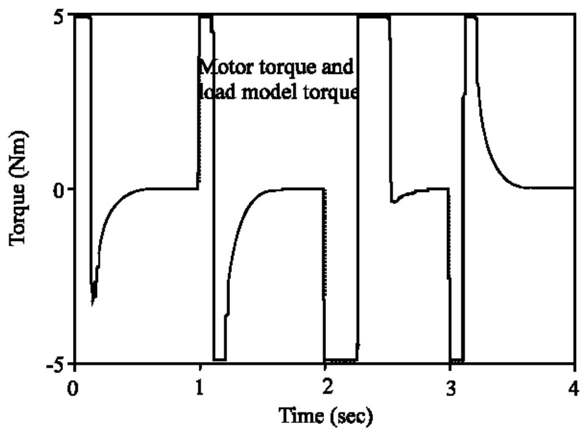

Fig. 4: The motor and dynamometer torques for the load model given by Eq. 13

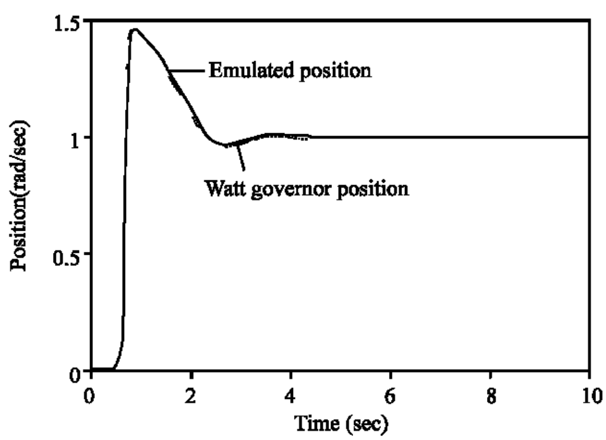

Fig. 5: The load model and the emulated position for the load model given by Eq. 14

learning algorithm with the learning rate of $2*10^{-3}$. Firstly, one link robot arm model given in Eq. 13 is emulated and results given in Fig. 3.

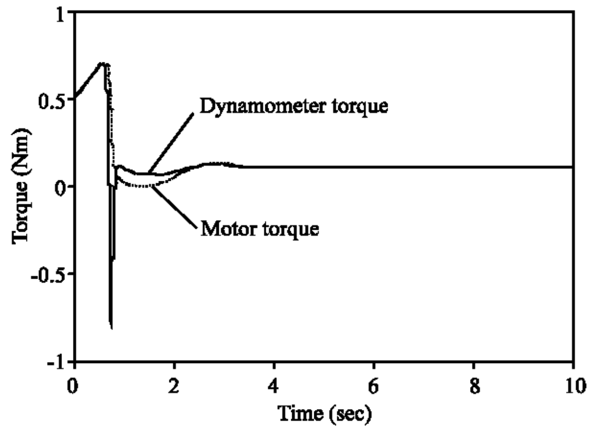

Fig. 6: The motor and dynamometer torques for the load model given by Eq. 14

$$T_e = (J + ml^2)\frac{d}{dt}\omega_m + B\omega_m + mglSin(\theta_m) \qquad (13)$$

Where, m is the mass and l is the length of the robot arm. Equation 16 implies that the drive machine is faced on the position dependent friction and mass and length dependent inertia. The load emulation performance of the trained NFC for the step reference input is tested for the load model of Eq. 13. The position error between the motor and load model position shows that the performance of the emulation. Thus the excellent emulation performance is obtained as shown in Fig. 3.

The torque applied to the motor and the load model and the load machine torque produced by the NFC to emulate the load of Eq. 13 is shown in Fig. 4.

The Watt governor is a good example of a non-linear load having an effective inertia and friction variation during motion[5,6]. It consists of two pendulums fixed at motor shaft; when the shaft rotates, the balls fly outwards due to the centrifugal force.

$$\begin{aligned}
\dot{x}_1 &= -\frac{B + 2m\ell^2 x_2 \sin(2x_3)}{J + 2m\ell^2 \sin^2(x_3)} x_1 + \frac{1}{J + 2m\ell^2 \sin^2(x_3)} T_e \\
\dot{x}_2 &= -\frac{B_o}{m\ell^2} x_2 + \frac{1}{2} x_1^2 \sin(2x_3) - \frac{g}{\ell}\sin(x_3) \qquad (14) \\
\dot{x}_3 &= x_2
\end{aligned}$$

where, x_1 is the motor speed, x_2 is the balls speed and x_3 is the balls position. Mass and length of the balls are m and l, respectively. The negligible error between the emulated position and the actual watt governor position shows that the excellent emulation performance is obtained as given in Fig. 5.

The torque applied to the motor and the watt governor model and the dynamometer torque produced by the NFC to emulate the load of Eq. 14 is shown in Fig.6.

CONCLUSIONS

This study proposes a method for the neuro-fuzzy control of a dynamometer to emulate the dynamical loads. Simulation results showing the performance of the emulation strategy are presented. The proposed emulation approach provides a simple and effective way to emulate the various nonlinear mechanical loads.

REFERENCES

1. Newton, R.W., R.E. Betz and H.B. Penfold, 1995. Emulating dynamic load characteristics using a dynamic dynamometer. Intl. Conf. Power Electron. and Drive Systems, pp: 465-470.
2. Carrol, J.J., D.M. Dawson and E.R. Collins, 1993. A non-linear control technique for the development of a computer controlled dynamometer. Dynamic Sys. and Contr. Division, American Soc. Mech. Eng., pp: 31-36.
3. Sandholdt, P., E. Ritchie, J.K. Pederson and R.E. Betz, 1996. A dynamometer performing dynamical emulation of loads with non-linear friction. IEEE Intl. Symp. Ind. Elec., pp: 873-878.
4. Hewson, C.R., G.M. Asher and M. Sumner, 1998. Dynamometer control for emulation of mechanical loads. IEEE Intl. Conf. Elect. Mach. (ICEM), pp: 1511-1518.
5. Akpolat, Z.H., G.M. Asher and J.C. Clare, 1999. Dynamic emulation of mechanical loads using a vector controlled induction motor-generator set. IEEE Trans. Industrial Electronics, 46: 370-379.
6. Akpolat, Z.H., G.M. Asher and J.C. Clare, 1999. Experimental dynamometer emulation of non-linear mechanical loads. IEEE Trans. Industry Applications, 35: 1367-1373.
7. Jang, J.S.R., C.T. Sun and E. Mizutani, 1997. Neuro-Fuzzy and Soft Computing, Prentice Hall.
8. Lazerini, B., L.M. Reyneri and M. Chiaberge, 1999. A neuro-fuzzy Approach to hybrid intelligent control. IEEE Transactions on Industry Applications, 35: 413-425.
9. Lin, F.J. and R.J. Wai, 2001. Hybrid control using recurrent fuzzy neural networks for linear induction motor servo drive. IEEE Transactions on Fuzzy Systems, 9: 102-115.
10. Da, F. and W. Song, 2003. Fuzzy neural networks for direct adaptive control. IEEE Transactions on Industrial Electronics, 50: 507-513.
11. Rahman, M.A. and P. Zhow, 1996. Analysis of brushless permanent magnet synchronous motors. IEEE Transactions on Industrial Electronics, 43: 256-267.