Neuro-Fuzzy Control of a Dynamometer for the Emulation of Nonlinear Mechanical Loads

Muammer GÖKBULUT[*], Beşir DANDIL[**]
[*]Firat University, Faculty of Technical Education, Department of Electronics and Computer Science
[**] Firat University, Faculty of Technical Education, Department of Electrical Science

ABSTRACT

This paper proposes a neuro-fuzzy controller for a vector controlled load machine (dynamometer) which is mechanically coupled to a drive machine for the emulation of nonlinear loads. The main objective is to provide a system for testing of high performance electrical drives under nonlinear loads. The proposed emulation strategy is based on the model reference control approach using an on-line trained Neuro-Fuzzy Controller (NFC). The emulation is placed in the closed speed loop of the drive machine. Varieties of dynamic load models which are the nonlinear functions of the shaft speed are successfully emulated and the generalization capability of the trained NFC is shown. Simulation results showing the excellent performance of the proposed emulation strategy are presented.

Keywords: Neuro-fuzzy controller, mechanical load emulation, permanent magnet synchronous motor (PMSM).

Doğrusal Olmayan Mekanik Yüklerin Emulasyonu için Dinamometrelerin Sinirsel-Bulanık Denetimi

ÖZET

Bu makalede, doğrusal olmayan yüklerin emulasyonunu yapmak amacıyla bir elektrik motoruna mekanik kuplaj elemanı ile bağlanan vektör denetimli yük makinesinin (dinamometre) sinirsel bulanık ağlarla denetimi gerçekleştirilmiştir. Yük emulasyonun temel amacı, elektrik sürücü sistemlerini test etmek için yük makinesini denetlemek suretiyle arzu edilen bir yük elde etmektir. Önerilen emulasyon yönteminde, gerçek zamanlı olarak eğitilen Sinirsel Bulanık Denetleyiciler (SBD) kullanarak model referans denetim yaklaşımı kullanılmıştır. Sürücü makinenin kapalı çevrim hız denetim döngüsü içerisinde yük emulasyonu gerçekleştirilmiştir. Rotor hızının doğrusal olmayan fonksiyonları ile tanımlanan çeşitli dinamik yüklerin emulasyonu yapılmış ve eğitilmiş SBD' nin farklı yük modellerini genelleme yeteneği gösterilmiştir. Simulasyon sonuçları, önerilen emulasyon yönteminin performansının son derecede iyi olduğunu göstermektedir.

Anahtar Kelimeler: Sinirsel bulanık denetleyiciler (SBD), Mekanik yük emulasyonu, Kalıcı mıknatıslı senkron motorlar (KMSM).

1. INTRODUCTION

Load emulation involves the torque control of the load machine mechanically coupled to the drive machine such that, the drive machine will see a mechanical load equal to a desired load model. The load emulation can be used for the controller design and validation, the testing of electrical machines and variable speed drives under the complex load conditions, in the laboratory environment (1-4). Such a laboratory based emulation facility is often required when a load application is situated on a remote site or in a hazardous environment. Examples of these industrial applications include reciprocating pumps, escalators, large washing machines and compressors (5). Even if the parameters of such loads are approximately known, the ability to evaluate and test such applications off-site would be advantageous (6).

The main elements of the emulation are the dynamometer and its control system (7). In the early studies, dynamometers were often used to reproduce load with static characteristics. However, the emulation of dynamic loads has gained popularity since the most of the industrial mechanical loads are nonlinear and / or time varying. 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 (6).

The control strategies of the load machine may be different for the purpose of the emulation. However, the reasonable approach is to find the right load machine torque by minimization of the error between the desired load model speed and the actual speed. Some model reference load emulation strategies are developed to overcome the problems mentioned above. It is shown that the model reference load emulation strategies give good results for wide variety of linear and nonlinear loads (8). An extension of the model reference approach using the load torque estimation is also proposed (5). However, these emulation approaches are not adaptive and should be modified for time-varying and highly nonlinear mechanical loads. Design of modern controllers, such as sliding mode, adaptive, and robust controllers, usually involves complex mathematical analysis and, have many difficulties in controlling time-varying and nonlinear plants. This paper uses the model reference emulation strategy based on neuro-fuzzy control which is an adaptive and learning system in nature, to overcome the above difficulties. Thus, the inverse dynamic problems related to load emulation and complex mathematical design procedures for the other control techniques are eliminated.

In recent years, the number of new approaches using various neural network-based fuzzy systems has increased significantly (9-15). For example, recurrent fuzzy-neural networks control of linear induction motors (16), robot manipulator control using generalized fuzzy-neural networks (17) and, fuzzy neural networks-sliding mode control for direct adaptive control strategy (18), have been studied by researchers. Although, these studies show the effectiveness of the neural-fuzzy controllers for the electrical drives, to our best knowledge, the neuro-fuzzy control strategy is not applied to the emulation of nonlinear mechanical loads. In (19), neural network load emulation strategy is reported for various nonlinear loads. This paper presents the neuro-fuzzy control of the load machine which is directly coupled to the drive machine in order to emulate nonlinear mechanical loads. The emulation is placed in the closed loop speed control system of the drive machine and, the proposed emulation scheme is tested for varieties of nonlinear loads. The simulation results showing the effectiveness of the proposed emulation strategy for various nonlinear loads are presented.

2. CONVENTIONAL LOAD EMULATION STRATEGIES

For testing variable speed electrical drives, static load tests are not sufficient since the dynamic and/or nonlinear effects in some loads can be dominant. Therefore, a dynamometer for testing of electrical drives should be able to perform dynamical tests, i.e., the dynamometer controller should be designed to implement the $G_{em} = \omega / T_e$ relation (5). Where, ω is the shaft speed, T_e is the electrical torque of the drive machine and G_{em} is the emulated load dynamics. Two common approaches for the dynamic load emulation are shown in Figure 1.

Inverse model emulation strategy as shown in Figure 1.a uses the inverse load dynamics to determine 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 (6). In the model reference approach as shown in Figure 1.b, a conventional PI controller G_c is designed to maintain the machine shaft speed at the desired speed (emulated load speed). G_{cl} is the closed loop transfer function of the speed control loop and used for compensating of unwanted dynamics of the speed loop. It is shown that the load emulation strategy in Fig.1.b gives the excellent results for variety of linear and nonlinear loads (5,6,8). However, these emulation approaches are not adaptive and should be modified for time-varying and highly nonlinear mechanical loads. Neuro-fuzzy emulation strategy proposed in this paper is an adaptive and learning system based on model reference approach in Fig. 1.b, which is favorable for time-varying and highly nonlinear mechanical loads.

3. NEURO-FUZZY LOAD EMULATION FOR PMSM

The state space model of the permanent magnet synchronous motors referred to the rotor rotating reference frame is given by Eqn. (1):

$$\frac{d}{dt}\begin{bmatrix} i_q \\ i_d \end{bmatrix} = \begin{bmatrix} -\frac{R}{L_q} & 0 \\ 0 & -\frac{R}{L_d} \end{bmatrix}\begin{bmatrix} i_q \\ i_d \end{bmatrix} + \begin{bmatrix} \frac{1}{L_q} & 0 \\ 0 & \frac{1}{L_d} \end{bmatrix}\begin{bmatrix} v_q \\ v_d \end{bmatrix} + \begin{bmatrix} -\frac{\lambda}{L_q}\omega - \frac{L_d}{L_q}.\omega.i_d \\ \frac{L_q}{L_d}\omega.i_q \end{bmatrix} \quad (1)$$

where, i_d and i_q are the direct and quadrature components of the stator currents, R is the stator resistance, L_q and L_d are the inductance, λ is the magnitude of the flux linkage established by the rotor magnet and ω is the rotor speed. The model is based on the assumptions that the air-gap is uniform and the

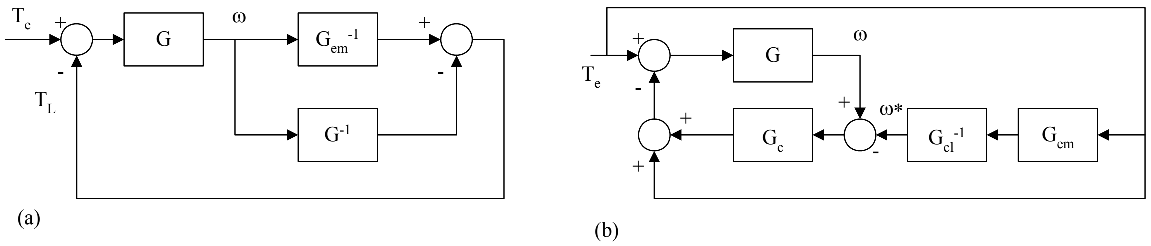

Fig. 1 a-) Load emulation using inverse load dynamics
b-) Model reference load emulation strategy

motor is supplied by three phase sinusoidal voltages (11,12). For the vector control of electrical drives, stator currents are decomposed into the flux and torque components which can be controlled independently. Using the flux and torque current components, electromagnetic torque produced by the PMSM is given by

$$T_e = K_T i_q + (L_d - L_q) i_q i_d \qquad (2)$$

where K_T is the torque constant. In the surface mounted permanent magnet synchronous motors, the d-axis inductance is assumed equal to the q-axis inductance. Thus, electromagnetic torque equation becomes dependent only on the q-axis current, i.e., $T_e = K_T i_q$. On the other hand, mechanical torque equation for electrical drives is given by

$$T_e - T_L = J \frac{d}{dt}\omega + B\omega \qquad (3)$$

where T_e is the drive machine torque (Nm), T_L is the load machine torque (Nm) and, total inertia and friction seen from the drive and the load machine for direct coupled system are $J=J_d+J_L$ and $B=B_d+B_L$ respectively. Thus, two control loops exist which are current and speed control loops. Hence, appropriate controllers should be designed for the drive machine speed and current loops and also for the load machine current loop since the aim of the load emulation is to control the load machine torque (or q-axis current) to provide a desired load model. This study focuses on the neuro-fuzzy controller design for the load machine. NFC uses the model following error and the change of error as inputs and produces the change of the reference q-axis current (torque current) for the load machine. Then, the reference q-axis current is obtained by integrating the NFC output in order to eliminate the steady state error. NFC is trained using the model following error propagated from the load machine. Block diagram of the proposed neuro-fuzzy load emulation system is given in Figure 2.

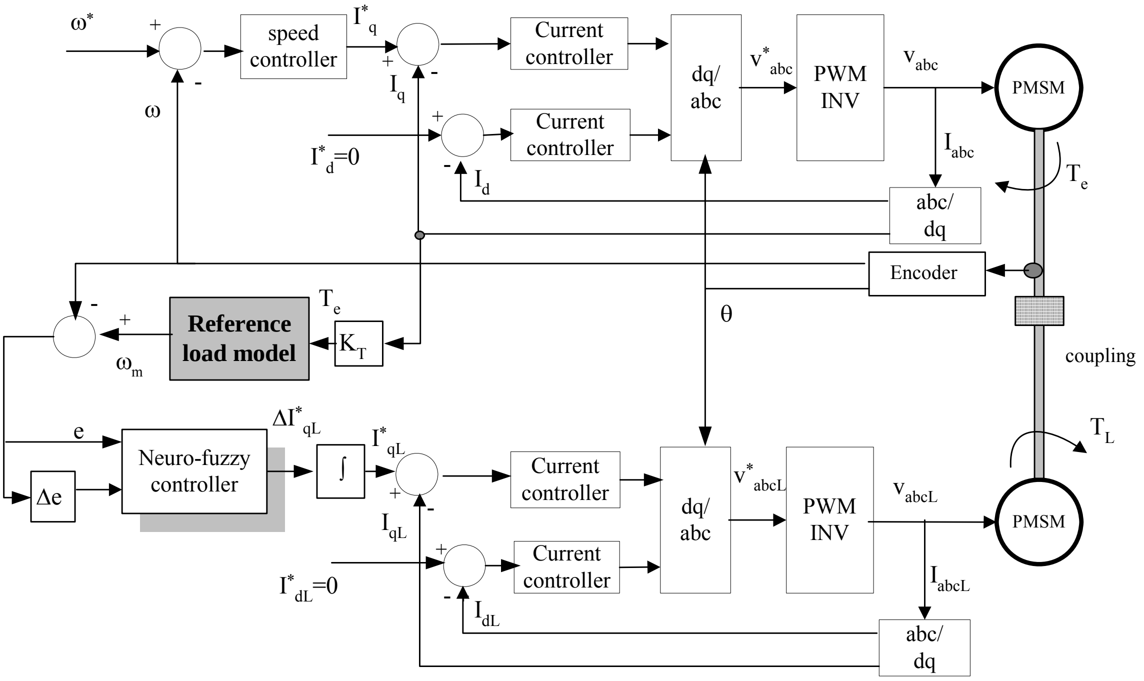

Fig. 2.Neuro-fuzzy Load Emulation Scheme.

With the implementation of field oriented control, PMSM emulation system shown in Fig.2 can be represented by the simplified block diagram as shown in Figure 3, in which the torque is $T_e = K_T i_q$ and plant transfer function is $G_p(s) = 1 /(Js + B)$.

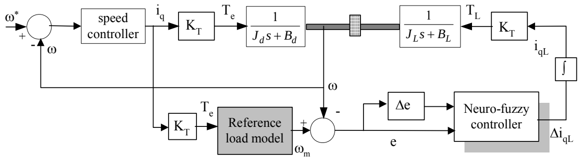

Fig. 3. Simplified block diagram of the neuro-fuzzy load emulation.

Let the inertia J_m and friction B_m of the reference load model (the load to be emulated) be a nonlinear function of the total inertia and speed, friction and speed respectively. The reference load model can be written as

$$T_e = J_m \frac{d}{dt}\omega_m + B_m\omega_m \qquad (4)$$

where ω_m is the reference load model speed. The purpose of the neuro-fuzzy load emulation is to design a neuro-fuzzy controller for the load machine to minimize the performance criteria depending on the speed error between the reference load model and motor.

$$e(t) = \omega_m(t) - \omega(t) \qquad (5)$$

This actually means that a good load emulation is obtained if the error *e(t)* is kept close to zero. Note that the type of the nonlinearity in the reference load model is not important 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 nonlinear function of the shaft inertia, friction and speed, i.e., $J_m = f(J, \omega, \dot{\omega})$ and $B_m = g(B, \omega, \dot{\omega})$.

4. NEURO-FUZZY CONTROLLERS

Some neuro-fuzzy architectures based on fuzzy inference method, selection of neuro-fuzzy inputs, and feed forward or feedback connections between the layers are discussed in (9,12,14). A five-layer Sugeno-type neuro-fuzzy system shown in Figure 4 is adopted in this study to implement the neuro-fuzzy controller. The neuro-fuzzy controller comprised of membership (y^1), rule (y^2), normalization (y^3), inference (y^4) and the output (y^o) layers. The model following error and the error change are chosen as the inputs for NFC. The change of the quadrature component of the desired dynamometer current is the output. The output of the neuro-fuzzy controller is integrated to eliminate the steady state error.

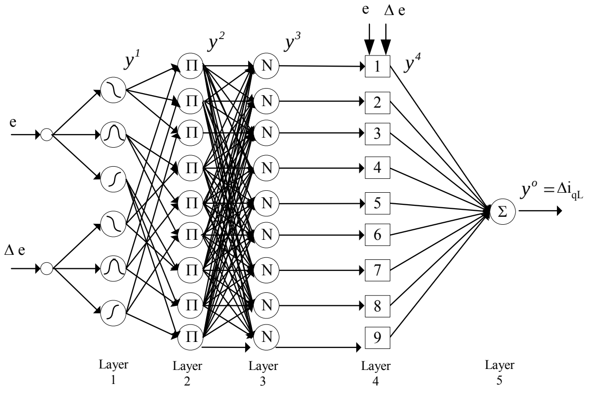

Fig. 4. Neuro-fuzzy controller

The first layer includes the membership functions and calculates the degree of membership functions for the input values. Three membership functions are used for each input. Sigmoid functions are adopted for the small and large values of the inputs, and generalized Bell functions are used for the other input values. Output of the first layer (y^1_{ri}: from input *r=1,2* to output *i=1,2,3*) is expressed as,

$$y^1_{ri} = \frac{1}{1 + e^{-a_{ri}(x_r - c_{ri})}} \quad i = 1,3$$

$$y^1_{r2} = \frac{1}{1 + \left| \dfrac{x_r - c_{r2}}{a_{r2}} \right|^{2b_{r2}}} \qquad (6)$$

where, a_{ri}, b_{ri} and c_{ri} are the parameters of the membership functions to be determined by training and NFC inputs are error and error change: $x_1 = e$ and $x_2 = \Delta e$. As the values of these parameters change, membership functions vary accordingly, thus exhibiting the most suitable forms of membership functions for fuzzy set. The second layer of the NFC includes the fuzzy rule base and the nodes in this layer represented by Π determine the fuzzy rules. The output of this layer which represents the firing strength of rules is given for three membership functions:

$$y^2_{ij} = y^1_{1i} . y^1_{2j} \qquad i = 1,2,3 \quad j = 1,2,3 \qquad (7)$$

The third layer is normalization layer labeled by N and calculates the ratio of the *ith* rule's firing strength to the sum of all rule's firing strength:

$$y^3_n = \frac{y^2_{ij}}{\sum_i \sum_j y^2_{ij}} \qquad n = 1,2,..........i * j \qquad (8)$$

The fourth layer which has adaptive nodes gives the certainty of rules. For the first-order Sugeno neuro-fuzzy systems (9), the output of this layer denoted by is the product of the normalized firing strength of layer 3 and an adaptive node function given by

$$y^4_n = y^3_n (p_n x_1 + q_n x_2 + r_n) \qquad (9)$$

where, p_n, q_n and r_n are the output function parameters of the NFC to be determined by the training of the NFC. The fifth layer is the output layer and produces an output as the sum of all incoming signals. Hence, the output of this layer

$$y^o = \Delta i_{qL} = \sum_n y^4_n \qquad (10)$$

is the change of the desired torque current for the load machine. From Eqn.(9) and Eqn.(10), the output of the NFC can be written as,

$$y^0 = \theta^T .X \qquad (11)$$

where, the NFC output parameter vector is $\theta^T = [p_1, q_1, r_1, ... \ p_n, q_n, r_n]$ and the regressor is

$$X = [y_1^3.x_1, y_1^3.x_2, y_1^3, ...y_n^3.x_1, y_n^3.x_2, y_n^3].$$

4.1. Training of the Neuro-Fuzzy Controller

NFC used for the load emulation of the vector controlled permanent magnet synchronous motor has two groups of parameters to be modified, which are the parameters of membership and the output functions as given in Eqn.(6) and (9). These parameters should be adapted in order to obtain the desired control performance. For the pattern learning algorithm, speed tracking error in discrete time is,

$$e(k) = \omega_m(k) - \omega(k) \qquad (12)$$

and the cost function to be minimized is,

$$E(k) = \frac{1}{2}e^2(k) \qquad (13)$$

where, $_m$ is the reference (load model) speed, is the actual shaft speed and *e* is the speed error. If θ is the parameter vector in Eqn. (11), i.e., $\theta = [\ p_1, q_1, r_1, ...p_n, q_n, r_n\ ]$, for the output of the NFC, then the back propagation learning algorithm can be defined as

$$\theta_i(k) = \theta_i(k-1) - \eta\left(\frac{\partial E(k)}{\partial \theta_i}\right) \qquad (14)$$

where the η *is* the learning rate. Hence, the gradient of the cost function can be written as

$$\frac{\partial E(k)}{\partial \theta_i} = \frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \omega(k)}\frac{\partial \omega(k)}{\partial \Delta i_{qL}(k)}\frac{\partial \Delta i_{qL}(k)}{\partial \theta_i} = -e(k)\frac{\partial \omega(k)}{\partial \Delta i_{qL}(k)}\frac{\partial \Delta i_{qL}(k)}{\partial \theta_i} \qquad (15)$$

where, $\dfrac{\partial \Delta i_{qL}}{\partial \theta_i} = X_i$ is easily calculated from the Eqns. (9-11). However, $\dfrac{\partial \omega}{\partial \Delta i_{qL}}$ should be calculated using the load machine dynamics.

The gradient of complex nonlinear dynamic systems cannot be found explicitly. Therefore, the approximate value of gradient should be used (i.e., the approximate value may be the discrete derivative of the system output with respect to the input, or sign of the discrete derivative). Hence, the local error δ at the output of the NFC is given as,

$$\delta = \frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \omega(k)}\frac{\partial \omega(k)}{\partial \Delta i_{qL}(k)} = -e(k)sgn(\frac{\omega(k) - \omega(k-1)}{\Delta i_{qL}(k) - \Delta i_{qL}(k-1)}) \qquad (16)$$

In this study, delta adaptation rule proposed in (10), is given by

$$\delta = Ae + \dot{e} \qquad (17)$$

which eliminates the calculation of the gradient of load machines. In a similar way, if the parameter vector *W* of the membership functions is defined as $W = [\ a_{ri}, b_{ri}, c_{ri}\ ]$, the gradient of the cost function for a parameter W_i of the membership functions can be written as,

$$\frac{\partial E(k)}{\partial W_i} = \delta \frac{\partial \Delta i_{qL}}{\partial {y_n}^4}\frac{\partial {y_n}^4}{\partial {y_n}^3}\frac{\partial {y_n}^3}{\partial {y_{ij}}^2}\frac{\partial {y_{ij}}^2}{\partial {y_{ri}}^1}\frac{\partial {y_{ri}}^1}{\partial W_i} \qquad (18)$$

Derivatives in Eqn.(18) can be calculated using the Eqn.(6-10).

4.2. Stability Analysis

If the parameters of the motor and load machine are explicitly known, from Eqn.(3) and Eqn.(4), the perfect control torque of the load machine can be derived as,

$$T_L^* = (J_m - J)\frac{d}{dt}\omega_m + (B_m - B)\omega_m \qquad (19)$$

Using this torque, the dynamic equation of the tracking error, $e = \omega - \omega_m$, can be expressed as

$$\dot{e} + \frac{B}{J}e = 0 \qquad (20)$$

and the tracking error *e* converges to zero. However, since external disturbances and unmodeled dynamics are unknown, the perfect control law cannot be implemented in practice. Therefore, a NFC is proposed to generate the desired load machine torque to approximate the perfect control law, i.e., $T_L^* = T_{NFC}^*$. Since the actual control law is $T_L = T_{NFC}$, using the Eqn.(3) for perfect and actual control law respectively, tracking error dynamics can be written as,

$$\dot{e} = -\frac{B}{J}e + \frac{1}{J}(T_{NFC}^* - T_{NFC}) \qquad (21)$$

where, $T_{NFC}^* = \theta^{*T} X^*$, $T_{NFC} = \theta^T X$ and, θ^T *and* X are defined in Eqn.(11). To proceed further, it can be assumed that, for each time step, the regressor *X* obtained from NFC is optimal (16,17,20) i.e., $X = X^*$. Thus, Eqn.(21) can be written in the following form.

$$\dot{e} = -\frac{B}{J}e + \frac{1}{J}((\theta^* - \theta)^T X) \qquad (22)$$

Now, let us consider the Lyapunov function candidate based on Eqn.(22) as

$$V(t) = \frac{1}{2}e^2 + \frac{1}{2}\lambda.tr\left[(\theta^* - \theta)^T.(\theta^* - \theta)\right]. \qquad (23)$$

By substituting Eqn. (22) in the derivative of Eqn. (23), the derivative of the Lyapunov function is obtained as

$$\dot{V}(t) = -\frac{B}{J}e^2 + \frac{1}{J}e.((\theta^* - \theta)^T.X) - \lambda.tr\left[(\theta^* - \theta)^T.\dot{\theta}\right] \qquad (24)$$

which is semi-definite, i.e., the control system will be asymptotically stable, if the adaptation law is chosen as

$$\dot{\theta} = \alpha.e.X\ . \qquad (25)$$

5. SIMULATION RESULTS AND DISCUSSION

The neuro-fuzzy load emulation system given in Fig.2 is simulated for the PMSM drive and dynamometer set which has the nominal parameters J=1*10^{-4} kgm^2, B=1*10^{-5} Nms, □=0.11 Wb/m^2,.R=11.5 ohm, L=0.0215 H, and P=6. The nuero-fuzzy controller is trained using pattern learning algorithm with the learning rate of 2*10^{-3}. Since the aim of this study is to control the load machine using the NFC to provide a desired load model for the drive machine, PI speed controller for the drive machine and the current controllers for the drive and load machine are not our main concern. In this section, some mechanical loads are emulated, and simulation results showing the emulation performance of the NFC are given.

Case 1: The emulation performance of the neuro-fuzzy controller is tested for the load model given by

$$T_e = (10J + K_j.\omega_m{}^2)\frac{d}{dt}\omega_m + (10B + B_a.\omega_m)\omega_m \qquad (26)$$

where the constants are K$_j$=5*10^{-9} and B$_a$=4*10^{-7}. Load model in Eqn.(26) implies that the drive machine is faced on the speed dependent inertia and friction.

when the load emulation performance of the trained NFC for the step reference input is tested for nonlinear load model of Eqn. (26), excellent model tracking performance is observed and shown in Figure 5.

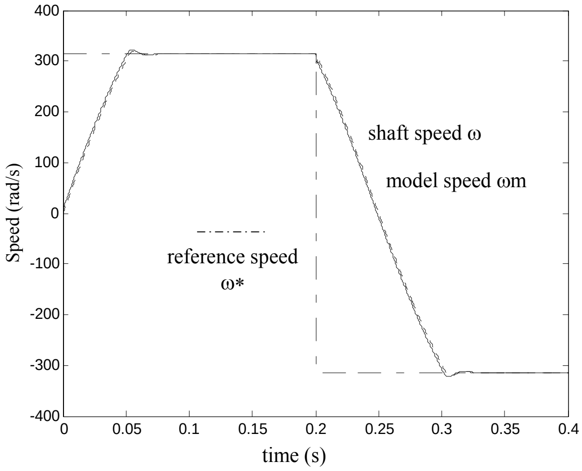

Fig. 5. The reference speed, model and the shaft speeds for the load model given by Eqn.(26).

As seen in Figure 5, the difference between the shaft and model speeds is very small, i.e., an excellent emulation performance is obtained. The drive machine torque applied to the motor and the load model and, the load machine torque produced by the NFC to emulate the load of Eqn.(26) is shown in Figure 6.

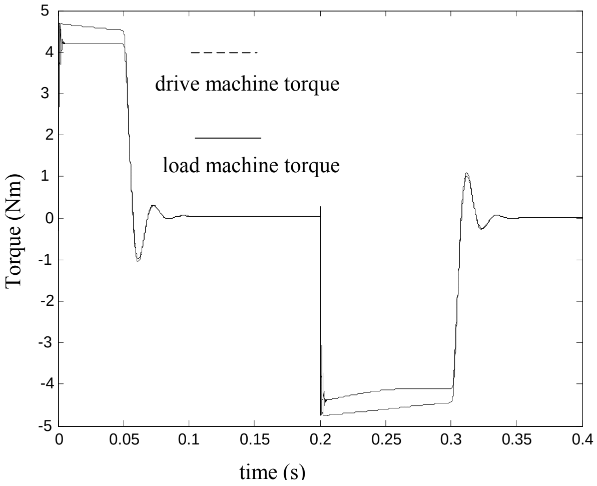

Fig. 6. The drive machine torque and the load machine torque produced by NFC.

Case 2: Emulation performance of the trained NFC is examined for the load model given by Eqn. (27), in which the emulated inertia and friction change as sinusoidal functions of the speed.

$$T_e = (4J + 3JSin(0.02\omega_m))\frac{d}{dt}\omega_m + (10B + 5BCos(0.02\omega_m))\omega_m \qquad (27)$$

Emulation performance of the trained NFC for the step reference is shown in Figure 7 for the load model given by Eqn. (27). As seen in Figure 7, the negligible error between the shaft and model speed implies that the emulation of load model given by (27) is excellent.

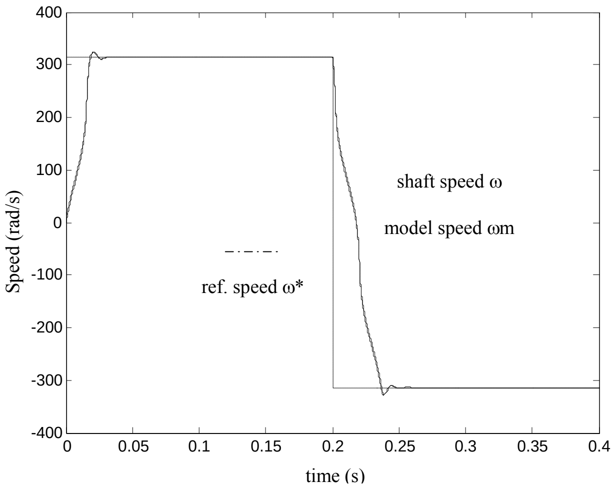

Fig. 7. Model and shaft speeds for the load model given by Eqn.(27)

Case 3: The Watt governor is a good example of a physical non-linear load having an effective inertia and friction variations during the motion. It consists of two pendulums fixed at O; when the shaft rotates, the balls fly outwards due to the centrifugal force.

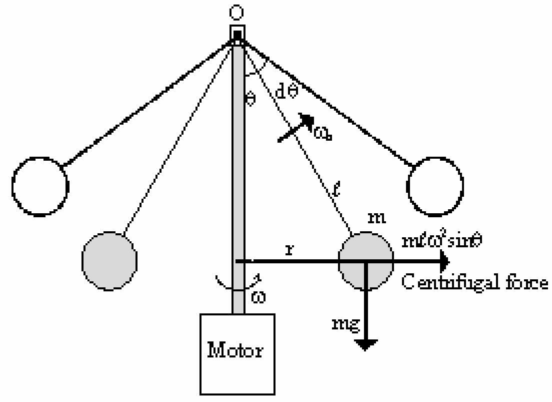

Fig. 8. The physical structure of a Watt governor

Defining the states as $\underline{x} = [\omega, \dot{\theta}, \theta]^T$, the state equations can be derived as (6,8)

$$\begin{aligned} \dot{x}_1 &= -\frac{B + 2m\ell^2 x_2 \sin(2x_3)}{J + 2m\ell^2 \sin^2(x_3)} x_1 + \frac{1}{J + 2m\ell^2 \sin^2(x_3)} T_e \\ \dot{x}_2 &= -\frac{B_o}{m\ell^2} x_2 + \frac{1}{2} x_1^2 \sin(2x_3) - \frac{g}{\ell} \sin(x_3) \\ \dot{x}_3 &= x_2 \end{aligned} \tag{28}$$

where, x_1, x_2 *and* x_3 are the motor speed, balls speed and balls position respectively. *m* is the mass of the balls and *l* is the length of the arms holding the balls. Neuro-fuzzy load emulation scheme is tested for the Watt governor model and emulation performance for the step reference is given in Fig. 9. Although an effective inertia and friction variation occurs during the motion, good emulation performance is obtained again.

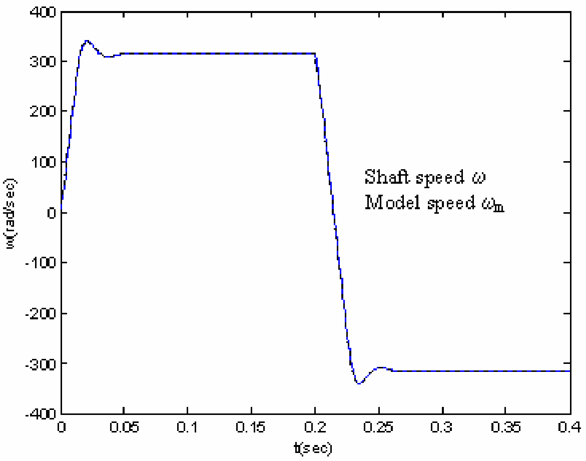

Fig. 9. Model and shaft speeds for the load model of Watt governor.

Real time implementation of the proposed neuro-fuzzy load emulation scheme requires an advanced and high speed digital signal processor to implement the control algorithms synchronously for both drive and load machine. Simulation results given above validate the proposed emulation scheme. Hence, real time training of the NFC may not be required when the off line trained NFC is used in experiments. However, real time training of NFC may be necessary, due to the uncertainties of the plant, and this may results in high gains the for the neuro-fuzzy controller, and the high gains cause noise problems in control systems. Thus the bandwidth of the emulation is usually limited by the noise in practice.

6. CONCLUSION

In this paper, a new load emulation technique based on the neuro-fuzzy control strategy is proposed. Comparing to the other control techniques, this emulation strategy has an adaptive and learning system in nature, and do not need the complex mathematical design procedure. Simulation results showing the performance of the emulation strategy are presented for various mechanical loads. The proposed emulation approach provides a simple and effective way to emulate the time varying and nonlinear industrial loads. Real time implementation of this emulation technique will help to test the control algorithms of the electrical drives under wide varieties of industrial loads in a laboratory environment.

7. REFERENCES

1. R.W. Newton, R.E.Betz, H.B. Penfold, Emulating dynamic load characteristics using a dynamic dynamometer, Proc. Int. Conf. Power Electron. and Drive Systems, 1995, 465-470.

2. R.E. Betz, H.B. Penfold, R.W. Newton, Local vector control of an AC drive system load simulator, Proc. IEEE Conf. Contr. Appl, 1994, 721-726.

3. J.J. Carrol, D.M. Dawson, E.R. Collins, A non-linear control technique for the development of a computer

controlled dynamometer, Proc. Dynamic Sys. and Contr. Division, American Soc. of Mech. Eng., 1993, 31-36.

4. P. Sandholdt, E. Ritchie, J.K. Pederson, R.E. Betz, A Dynamometer Performing Dynamical Emulation of Loads with Non-Linear Friction, Proc. IEEE int. Symp. Ind. Elec. 1996, 873-878.

5. C.R. Hewson, G.M. Asher, M. Sumner, Dynamometer Control for Emulation of Mechanical loads, Proc. IEEE Int. Conf. Elect. Mach. (ICEM), 1998, 1511-1518.

6. Z.H. Akpolat, G.M. Asher, J.C. Clare, Dynamic emulation of mechanical loads using a vector controlled induction motor-generator set, IEEE Trans. on Industrial Electronics, 46, 1999, 370-379.

7. B. Hassania, P. Sicard, A. Ba-razzouk, Solutions to Typical Load Emulation Control Problems, Proc. Electrimacs, 7. Int. Conf. Model. Simul. Elect. Mach., 2002.

8. Z.H. Akpolat, G.M. Asher, J.C. Clare, Experimental dynamometer emulation of non-linear mechanical loads, IEEE Trans. on Industry Applications, 35, (1), 1999, 1367-1373.

9. J-SR. Jang, C-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, (Prentice Hall,1997)

10. Y.C. Chen, C.C. Teng, A model reference control structure using a fuzzy neural network, Fuzzy Sets and Systems, 73, 1995,291-312.

11. M.A. Rahman, P. Zhow, Analysis of brushless permanent magnet synchronous motors, IEEE Transactions on Industrial Electronics, 43, 1996, 256-267.

12. F.J. Lin, R.J. Wai, H.P. Chen, A PM synchronous servo motor drive with an on-line trained fuzzy neural network controller, IEEE Transactions on Energy Conversion, 13,1998, 319-325.

13. C.H. Lee, C.C. Teng, Identification and control of dynamic systems using recurrent fuzzy neural Networks, IEEE Transactions on Fuzzy Systems, 8,2000,349-366.

14. B. Lazerini, L.M. Reyneri, M. Chiaberge, A neuro-fuzzy Approach to hybrid intelligent control, IEEE Transactions on Industry Applications, 35, 1999,413-425.

15. N. Barsoum, Artificial neuron controller for DC drive, Proc. Power Engineering Society Winter Meeting, 2000, 398 -402.

16. F. J. Lin and R. J. Wai, Hybrid control using recurrent fuzzy neural networks for linear induction motor servo drive, IEEE Transactions on Fuzzy systems, 9, 2001,102-115.

17. M J. Er and Y. Gao, Robust adaptive control of robot manipulators using generalized fuzzy neural networks, IEEE Transactions on Industrial Electronics, 50, 2003, 620-628.

18. F. Da and W. Song, Fuzzy neural networks for direct adaptive control, IEEE Transactions on Industrial Electronics, 50, 2003, 507-513.

19. M. Gokbulut, Z.H. Akpolat, H. Güldemir, “Emulation of nonlinear mechanical loads using multi-layer neural networks”, International Journal for Engineering Modelling, 13, 2000, 69-75.

20. (K. J. aström and B. Wittenmark, Adaptive control, Addision-Wesley, New York,1995.