LuGre Model-Based Neural Network Friction Compensator in a Linear Motor Stage

Rong-Hwang Horng[1], **Li-Ren Lin**[1] and **An-Chen Lee**[2,#]

1 Department of Mechanical Engineering, Graduate school, National Chiao-Tung University, Hsinchu, Taiwan, ROC
2 Department of Mechanical Engineering, National Chiao-Tung University, Hsinchu, Taiwan, ROC
Corresponding Author / E-mail:aclee@mail.nctu.edu.tw, TEL: +886-3-5712121 ext. 55106, FAX: +886-3-5725372

KEYWORDS : Linear-Motor-Driven Motion System, LuGre Friction Model, LuGre Model-Based Neural Network, Friction compensation

This paper proposes a LuGre Model-Based Neural Network (MBNN) friction compensation algorithm for a linear motor stage. For matching the friction phenomena in both the motion-start region and the motion-reverse region, the LuGre dynamic model is employed into the proposed compensation algorithm. After training of the model-based neural network is completed, the estimated friction for compensation is obtained. From the obtained result we find that the new structure gains advantage over the non-friction compensation system on the performance of the compensator in both regions. The proposed compensator is evaluated and compared experimentally with an uncompensated system on a microcomputer controlled linear motor tracking system in the final section of the paper. The experimental results show the improvement on the maximum velocity error and the root mean square tracking error in the motion-start region ranges from 34% to 53% and from 53% to 75% respectively, and in the motion-reverse region from 48% to 65% and from 79% to 90% respectively.

Manuscript received: May 23, 2005 / Accepted: September 12, 2005

1. Introduction

Friction is one of the major limitations in performing high precision motion systems. It induces undesirable phenomena such as stick-slip oscillation, steady state error, and poor tracking performance. For reducing the undesirable phenomena induced by friction, friction models for different fields, i.e. tribology, dynamics, and control[1, 2], are therefore developed. In general, frictions are grouped into two types, i.e., the static friction model and the dynamic friction model. While the static model defines static map between velocity and friction force which has static, coulomb, and viscous friction components, the dynamic friction model predicts the nonlinear behavior of friction under micro-dynamic scale and the macro-dynamic scale. Having a good dynamic friction model is a necessity for suitable control scheme to fulfill the stringent requirement in tracking and positioning accuracy.

Leonardo da Vinci, Coulomb, Renolds and Stribeck *et al.* made a contribution to construct various classical friction models[1]. They specified that the friction force at zero velocity should be equal to the sum of external force. It is hard to detect the velocity reversal point in simulations or experiments, however. Karnopp[3] proposed a method to predict friction without accurate detection of zero velocity. By Sepehri et al.[4], the Karnopp model was found possible to cause numerical instabilities in the stick phase. Leine[5] proposed a switch friction model to overcome the numerical instabilities. Unfortunately, the classical static models as above failed to provide any information about presliding displacement (micro-slip) in stiction regime and friction lag. Therefore, Armstrong-Helouvry[6] proposed a seven-parameter integrated friction model with various observed friction phenomena. This model comprises parameters to account for the presliding displacement, Coulomb, viscous, and Stribeck friction. The behavior of state variable friction models proposed by Armstrong-Helouvry resembles the behavior of a stiff (nonlinear) spring in pre-sliding region, and also describes the behaviors of slipping region without switch mechanism. Dahl model[7], which could predict friction lag between velocity reversals and leaded to hysteresis loops, was the first model in the form of state variable. Canudas de Wit *et al.*[8] extended and modified the Dahl model by including arbitrary steady state characteristics, and proposed a modified Dahl model (or the LuGre friction model). This model captured most of the friction behavior observed experimentally.

As motioned above, several friction models have been widely studied. Various compensation strategies based on different models were employed. Gao *et al.*[9] and Kovacic *et al.*[10] proposed black box neural networks to predict real friction for a DC motor servo system. They used the reference command, motion position, and velocity as the neural networks inputs. Gao *et al.*[9] performed an off-line training method to get the neural network weightings and only simulation results were shown in their research to compare their friction compensation performance. Kovacic *et al.*[10] constructed two neural networks to estimate static friction and gravitation-dependent load. They trained the first neural network from a Tustin's static model. Simultaneously, the second neural network has been trained from a shaft load with a regular sine function. The experimental results obtained showed that addition of neural network-based estimators may considerably increase the performance of the servo system. However, since the use of static fiction and the limit of sensor resolution, the approach can not capture the friction phenomenon in micro motion, such as the stick-slip and Stribeck effect. Furthermore, other nonlinear effects, such as the backlash, affected the experiment results. Yen *et al.*[11] combined two friction models, a black box neural network friction estimator for the direction-change region and a conventional Coulomb-friction-based input-output model for the non-direction-change region. From their experimental results, the tracking performance can be improved only in the direction-change region. It indicates that another neural network structure or friction compensator should be concerned in non-direction-change region to improve all tracking performance.

Black box neural networks provide only static mapping between inputs and outputs. Regardless of their type, however, neural networks are generally disadvantaged by their "black box" format. Furthermore, these networks require prohibitively extensive training, and are hard to interpret once trained[12]. Gan and Danai[12] proposed the model-based neural network (MBNN) for modeling nonlinear dynamic systems. The MBNN structure is formulated according to a linearized state space model of the dynamic system. After training weightings, adaptation to nonlinearities of the plant is confined to the activation functions of individual nodes. In this paper, the concept of model-based neural network is adopted for nonlinear dynamic friction modelling based on the known friction model, the LuGre friction model. We use multilayer neural networks (MNN) method to perform the LuGre MBNN. The neural network weightings are different in motion start region and in velocity reversal region since the friction behavior is different in these two regions[13]. Suitable initial training coefficients and training signals are chosen to update the neural network weightings in each region. Furthermore, we combine two neural networks in each region for friction compensation in a linear slide motion control system. The remainder of this paper is organized as follows. In the following section, the experimental system and the LuGre friction model are described. In section 3, a LuGre model-based neural network friction compensator and the back-propagation training procedures are proposed. Comparisons of the experimental results in motion start region and the velocity reversal region are made in section 4. Conclusions are given in section 5.

2. Dynamics of Linear Motor Stage

The linear slide systems are the most common applications of motion control. Traditionally, most of the linear slide systems are ball-screw-driven, but the linear-motor-driven systems are becoming popular in recent years due to their simple structure and absence of flexible coupling. From the friction study viewpoint, the existing backlash and compliance in a ball-screw-driven system may induce nonlinear phenomena with multi-source friction effects. This makes it practically impossible to distinguish friction from other nonlinear effects. On the contrary, linear-motor-driven systems are free from the complicated situation because nonlinear backlash and multi-source frictions do not exist in the systems. The observed friction behaviors will be quite different for the same reason. In this paper, a linear-motor-driven motion system is under study.

2.1 Hardware setup

The experimental motion system illustrated in Fig. 1 and Fig. 2 consists of following components: a linear-motor-driven motion system, a laser displacement meter, and a PC (PC1 in Fig. 1) with a DAC and encoder interface. The linear motor system is composed of a linear motor (IL6-050A1) and an AC servo amplifier (SERVOSTAR CD) operating in torque (current) mode, both of which are made by Kollmorgen Corporation[14]. In addition, two sensors are in use in this system, i.e., a linear scale (RENISHAW RGH24Y, resolution 0.1 micrometer) which provides position information for the vector control of servo amplifier, and a fiber optic laser encoder (RENISHAW RLE10) for measuring the displacement of the motion table with adjustable resolutions. The accuracy of the resolution supplied by RLE10 is influenced by environmental effects such as relative humidity, temperature, pressure and cosine errors. Hence, calibration by another measurement instrument is required. The RENISHOW laser interferometer system which includes an environmental compensation unit (EC10) is in use. After calibration, the basic length units (BLU) for course and fine resolutions are found to be 0.0791μm and 0.020μm, respectively. The choice of resolution scale is dependent on the encoder transition time (1 MHz in our system), desired maximum velocity, and travel range.

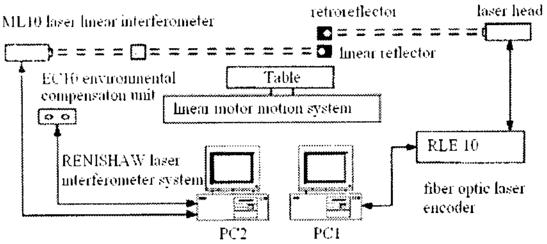
ML10 laser linear interferometer
retroreflector
laser head
linear reflector
EC10 environmental compensaton unit
Table
linear motor motion system
RLE 10
RENISHAW laser interferometer system
fiber optic laser encoder
PC2
PC1

Fig.1 The experimental linear-motor-driven motion system together with the resolution calibration system

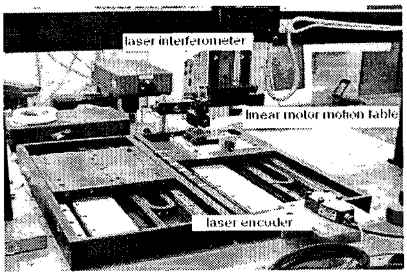

Fig.2 The photo of linear-motor-driven motion stage

2.2 Modeling of the linear motor stage

In general, the bandwidth of current loop is very fast as compared with mechanical system. If the high frequency modes are ignored, the system equation can be simplified as

$$J\ddot{x}+B\dot{x}+T_f=u \tag{1}$$

where J is the inertia (equivalent mass); $B\dot{x}+T_f$ is the friction force; and u is the input force to the system generated by a current-controlled servo amplifier with PI velocity loop controller and P position loop controller. The damping coefficient is considered as a parameter of the controlled plant in this paper, therefore T_f represents the friction force without viscous friction term. Generally, T_f is function of position, velocity and control input force.

2.3 The dynamic LuGre model

In this section, the LuGre model is briefly described first. The interface of two contact rigid bodies can be modeled as a lot of elastic bristles. When a tangential force is applied, the bristles will deflect like springs which give rise to friction force. If the force is sufficiently large, some of the bristles deflect so much that they will slip. The LuGre model based on the average behavior of the bristles is described as follows.

$$\frac{dz}{dt}=v-\frac{|v|}{g(v)}\sigma_0 z \tag{2}$$

$$T_f=\sigma_0 z+\sigma_1\frac{dz}{dt} \tag{3}$$

where z is the average bending displacement of bristles; v is the relative velocity between the two bodies; $g(v)$ is a positive function of velocity; σ_0 and σ_1 are the stiffness and damping coefficient of average behavior of bristles, and T_f is the friction force due to bristles' deflection. A term accounting for the viscous friction could be added to last equation, and the whole friction force becomes

$$f=\sigma_0 z+\sigma_1\frac{dz}{dt}+Bv \tag{4}$$

where B is the viscous friction coefficient.

The function $g(v)$ can be obtained by measuring the steady state friction when the velocity is held constant. When velocity reaches steady state, friction force is a static map versus velocity. Equations (5) and (6) show this situation.

$$f_{ss}=\sigma_0 z_{ss}+Bv \tag{5}$$

$$g(v) = F_S - (F_S - F_C)(1 - e^{-\left|\frac{v}{v_s}\right|}) \qquad (6)$$

where F_C is the Coulomb (kinetic) friction, F_S is the stick force, and the constant v_s is the Stribeck velocity. The detailed procedures for identifying parameters are described in Ref. 13 and the results of the underlying system are listed in Table 1.

Table 1 Parameters of the Experimental System

Symbol and name	Value and unit
J, inertia	5.37 Kg
B, viscous friction coefficient	111.65 Kg/s
F_S, Stick force	21.49N
F_C, Coulomb friction	13.02N
v_s, Stribeck velocity constant	5.01mm/s
σ_0, Bristle stiffness	$2.61\times10^6 Kg/s^2$
σ_1, Bristle damping Coeffi.	$1.28\times10^4 Kg/s$
K_t, force constant of motor	28.5 N/A
K_a, gain of current driver	0.349 Volt/A
T, sampling rate	0.0005 sec

From our previous research, no Stribeck friction appears near the velocity reverse region. Therefore, the positive function $g(v)$ in the LuGre model has to be modified as to let

$g(v) = F_C$

after the moving stage experiences Stribeck friction at motion start.[13]

3. LuGre Model-Based Neural Network Friction Compensator

3.1 Model-based neural network[12]

The neural network is structured according to a linearized state-space model of the underlying system. If the discrete-time nonlinear state-space model of the plant is defined as

$$x(k+1) = f(x(k), u(k)) \qquad (7)$$

$$y(k) = h(x(k), u(k)) \qquad (8)$$

where u(k), x(k) and y(k) represent the sampled values of the inputs, states, and output at time k, respectively, then the linearized state-space model has the form

$$x(k+1) = Ax(k) + Bu(k) \qquad (9)$$

$$y(k) = Cx(k) + Du(k) \qquad (10)$$

The above model can be formulated as a neural network, having the same number of input and outputs. An example of such a neural network is shown in Fig. 3 for a second-order model

$$\begin{Bmatrix} x_1(k+1) \\ x_2(k+1) \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} x_1(k) \\ x_2(k) \end{Bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(k) \qquad (11)$$

$$\begin{Bmatrix} y_1(k) \\ y_2(k) \end{Bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{Bmatrix} x_1(k) \\ x_2(k) \end{Bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} u(k) \qquad (12)$$

The functions *f*, *g*, *h* and *l*, shown in Fig. 3, are adjustable neural network functions.

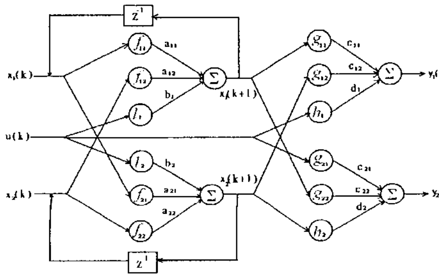

Fig.3 Model-based neural network representing a second-order plant[12]

3.2 LuGre Model-Based Neural Network

The LuGre model-based neural network is used to compensate the motion friction based on the LuGre friction model in Eq. (2) and (3). After substituting Eq. (2) into Eq. (3), we get that the discrete formula of Eq. (3) has the form

$$\begin{aligned} T_f(k) &= \sigma_0\left(1 - \frac{|v(k)|}{g(v(k))}\sigma_1\right)z(k) + \sigma_1 v(k) \qquad (13) \\ &= \phi_{nn}(v(k))z(k) + \sigma_1 v(k) \end{aligned}$$

where

$$\phi_{nn}(v(k)) = \sigma_0\left(1 - \frac{|v(k)|}{g(v(k))}\sigma_1\right) \qquad (14)$$

And Eq. (2) can be discretized by bilinear transformation

$$Z(k) = \varphi_{nn}(v(k), v(k-1))z(k-1) + \theta_{nn}(v(k))(v(k) + v(k-1)) \qquad (15)$$

where

$$\varphi_{nn}(v(k), v(k-1)) = \frac{1 - \frac{T}{2}\frac{|v(k-1)|}{g(v(k-1))}\sigma_0}{1 + \frac{T}{2}\frac{|v(k)|}{g(v(k))}\sigma_0} \qquad (16)$$

$$\theta_{nn}(v(k)) = \frac{\frac{T}{2}}{1 + \frac{T}{2}\frac{|v(k)|}{g(v(k))}\sigma_0} \qquad (17)$$

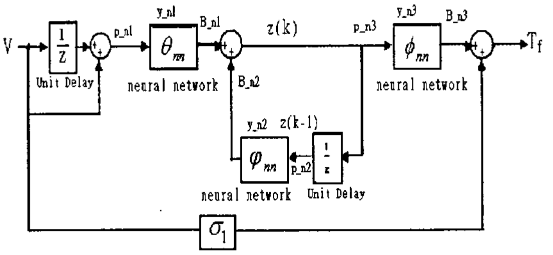

Fig. 4 LuGre Model-Based Neural Network (MBNN) structure

Then, we can get the block diagram of model-based neural network structure, shown in Fig. 4, by Eq. (13) and Eq. (15). In this structure, θ_{nn}, φ_{nn} and ϕ_{nn} are the neural networks and the corresponding outputs are *y_n1*, *y_n2* and *y_n3*. The θ_{nn} neural network has one input layer, one hidden layer with n neurons and one output layer. The φ_{nn} neural network has one input layer, one hidden layer with m neurons and one output layer. The ϕ_{nn} neural network has one input layer, one hidden layer with r neurons and one output layer. Then, the neural network outputs have the forms

$$\begin{aligned} y\_n1 &= \theta_{nn}(v(k)) \qquad (18) \\ &= \sum_{j=1}^{n} LW1_{j1} S1_j (IW1_{ij} \times u\_n1 + bi_j\_n1) + bo\_n1 \end{aligned}$$

$$\begin{aligned} y\_n2 &= \varphi_{nn}(v(k), v(k-1)) \qquad (19) \\ &= \sum_{j=1}^{m} LW2_{j1} S2_j (\sum_{i=1}^{2} IW2_{ij} \times u\_n2_i + bi_j\_n2) + bo\_n2 \end{aligned}$$

$$\begin{aligned} y\_n3 &= \phi_{nn}(v(k)) \qquad (20) \\ &= \sum_{j=1}^{r} LW3_{j1} S3_j (IW3_{ij} \times u\_n3 + bi_j\_n3) + bo\_n3 \end{aligned}$$

where

$S1_j$	activated function in θ_{nn} ;
$IW1_{ij}$	weighting from i-th input to j-th neuron in θ_{nn} ;
$LW1_{jk}$	weighting from j-th neuron to k-th output in θ_{nn} ;
bi_j_n1	bias of j-th hidden layer's neural in θ_{nn} ;
bo_n1	bias of output layer in θ_{nn} ;
u_n1	input in θ_{nn} , i.e. *v(k)*;
$S2_j$	activated function in φ_{nn} ;
$IW2_{ij}$	weighting from i-th input to j-th neural for φ_{nn} ;
$LW2_{jk}$	weighting from j-th neural to k-th output for φ_{nn} ;

bi_j_n2 bias of j-th hidden layer's neural in φ_{nn};
bo_n2 bias of output layer in φ_{nn};
u_n2_i input in φ_{nn}, i.e. *v(k), v(k-1)*;
$S3_j$ activated function in ϕ_{nn};
$IW3_{ij}$ weighting from i-th input to j-th neural for ϕ_{nn};
$LW3_{jk}$ weighting from j-th neural to k-th output for ϕ_{nn};
bi_j_n3 bias of j-th hidden layer's neural in ϕ_{nn};
bo_n3 bias of output layer in ϕ_{nn};
u_n3 input in ϕ_{nn}, i.e. *v(k)*.

The output of function blocks corresponding to θ_{nn}, φ_{nn} and ϕ_{nn} are

$$B\_n1 = p\_n1 \times y\_n1 \tag{21}$$

$$B\_n2 = p\_n2 \times y\_n2 \tag{22}$$

$$B\_n3 = p\_n3 \times y\_n3 \tag{23}$$

And the friction output of the LuGre MBNN structure is

$$T_f = B\_n3 + \sigma_1 v \tag{24}$$

3.3 Learning procedures for the LuGre MBNN friction compensator

In Fig. 4, the input and output of the LuGre MBNN are velocity and friction, respectively. The velocity is estimated from the fiber optic laser encoder by $\alpha\beta$ – filter[15] in our motion system. Since the friction is impossible to measure on-line and might change with environmental effects, such as humidity or temperature, the neural network weightings in Eq. (18) ~ (20) will be trained by using the learning architecture as shown in Fig. 5, where variant velocity commands are fed into the real plant and the nominal plant to train the neural network weightings. The manipulations in the dotted-line box in Fig. 5 were executed in a computer. The nominal plant, $1/(Js+B)$, in which the parameters *J* and *B* given in Table 1, is also modeled by MBNN. The estimated friction will approximate the real friction when network weighting approaches the optimal ones, i.e. the cost function reaches a small pre-designated value.

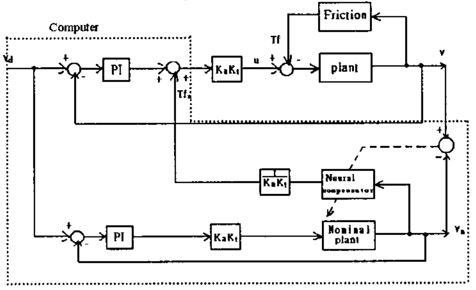

Fig. 5 Learning architecture for the LuGre MBNN friction compensator

In this study, we define the cost function as

$$E = \frac{1}{2}(v_n - v)^2 = \frac{1}{2}e^2 \tag{25}$$

Then, the partial differential equations of cost function with weightings and biases are

$$\frac{\partial E}{\partial LW1_{j1}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n1}\frac{\partial B\_n1}{\partial y\_n1}\frac{\partial y\_n1}{\partial LW1_{j1}} \tag{26}$$
$$= -e \cdot y\_n3 \cdot p\_n1 \cdot S1_j$$

$$\frac{\partial E}{\partial IW1_{ij}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n1}\frac{\partial B\_n1}{\partial y\_n1}\frac{\partial y\_n1}{\partial IW1_{ij}} \tag{27}$$
$$= -e \cdot y\_n3 \cdot p\_n1 \cdot LW1_{j1} \cdot \dot{S}1_j \cdot u\_n1_i$$

$$\frac{\partial E}{\partial bo\_n1} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n1}\frac{\partial B\_n1}{\partial y\_n1}\frac{\partial y\_n1}{\partial bo\_n1} \tag{28}$$
$$= -e \cdot y\_n3 \cdot p\_n1$$

$$\frac{\partial E}{\partial bi\_n1} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n1}\frac{\partial B\_n1}{\partial y\_n1}\frac{\partial y\_n1}{\partial bo\_n1} \tag{29}$$
$$= -e \cdot y\_n3 \cdot p\_n1 \cdot LW1_{j1} \cdot \dot{S}1_j$$

$$\frac{\partial E}{\partial LW2_{j1}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n2}\frac{\partial B\_n2}{\partial y\_n2}\frac{\partial y\_n2}{\partial LW2_{j1}} \tag{30}$$
$$= -e \cdot y\_n3 \cdot p\_n2 \cdot S2_j$$

$$\frac{\partial E}{\partial IW2_{ij}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n2}\frac{\partial B\_n2}{\partial y\_n2}\frac{\partial y\_n2}{\partial IW2_{ij}} \tag{31}$$
$$= -e \cdot y\_n3 \cdot p\_n2 \cdot LW2_{j1} \cdot \dot{S}2_j \cdot u\_n2_i$$

$$\frac{\partial E}{\partial bo\_n2} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n2}\frac{\partial B\_n2}{\partial y\_n2}\frac{\partial y\_n2}{\partial bo\_n2} \tag{32}$$
$$= -e \cdot y\_n3 \cdot p\_n2$$

$$\frac{\partial E}{\partial bi_j\_n2} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial Sn}\frac{\partial Sn}{\partial B\_n2}\frac{\partial B\_n2}{\partial y\_n2}\frac{\partial y\_n2}{\partial bi_j\_n2} \tag{33}$$
$$= -e \cdot y\_n3 \cdot p\_n2 \cdot LW2_{j1} \cdot \dot{S}2_j$$

$$\frac{\partial E}{\partial LW3_{j1}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial y\_n3}\frac{\partial y\_n3}{\partial LW3_{j1}} \tag{34}$$
$$= -e \cdot p\_n3 \cdot S3_j$$

$$\frac{\partial E}{\partial IW3_{ij}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial y\_n3}\frac{\partial y\_n3}{\partial IW3_{ij}} \tag{35}$$
$$= -e \cdot p\_n3 \cdot LW3_{j1} \cdot \dot{S}3_j \cdot u\_n3_i$$

$$\frac{\partial E}{\partial bo\_n3} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial y\_n3}\frac{\partial y\_n3}{\partial bo\_n3} \tag{36}$$
$$= -e \cdot p\_n3$$

$$\frac{\partial E}{\partial bi_j\_n3} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial v}\frac{\partial v}{\partial u}\frac{\partial u}{\partial Tf_n}\frac{\partial Tf_n}{\partial B\_n3}\frac{\partial B\_n3}{\partial y\_n3}\frac{\partial y\_n3}{\partial bi_j\_n3} \tag{37}$$
$$= -e \cdot p\_n3 \cdot LW3_{j1} \cdot \dot{S}3_j$$

We define the generalized weighting as *w*, then the optimal weighting will be obtained by the back propagation method[16]

$$w(k+1) = w(k) - \eta \cdot \nabla_w E(k) \tag{38}$$

where η is coefficient of learning speed and $\nabla_w E(k)$ is the gradient of cost function.

3.4 Control procedures for the LuGre MBNN friction compensator

After several training epochs are applied in each region, the optimal neural network weightings can be obtained as the cost function reaches a small pre-designated value. Then, the LuGre MBRNN friction compensator is applied to the motion system and the control architecture is shown in Fig. 6. Furthermore, in order to avoid the bouncing effect on the response, the compensated friction T_{fn} is designed as follows to make smooth transition in the control effort:

$$T_{fn} = \begin{cases} T_{f1} & \text{if } t < t_1 \\ (1-r)\cdot T_{f1} + r \cdot T_{f2} & \text{if } t_1 \le t \le t_2 \\ T_{f2} & \text{if } t_2 < t \end{cases} . \tag{39}$$

$$r = \frac{t - t_1}{t_2 - t_1} \tag{40}$$

where t_1 and t_2 are the switching times, T_{f1} and T_{f2} are the estimated frictions individually from the optimal neural network weightings in motion start region and in velocity reversal region, $r: 0 \le r \le 1$ is the switching parameter between time $t_1 \le t \le t_2$. The t_1 should be chosen after the moving stage leaves the Stribeck region while the choice of t_2 is not very critical and dependent on the practical applicaions since $T_{f1}(t_1) \approx T_{f2}(t_2)$ theoretically.

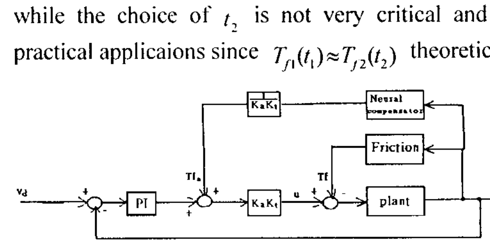

Fig. 6 LuGre MBNN friction compensator

4. Experimental Results

A linear motor feed drive servomechanism, shown in Fig. 1, is used to test the friction compensation performance of our proposed compensator. The LuGre friction dynamic model is learned to match different friction phenomena in both the motion-start region and the motion-reverse region.

In the LuGre MBNN structure, shown in Fig. 4, the neurons of hidden layer for θ_{nn}, φ_{nn} and ϕ_{nn} are chosen as ten and the neural activated function is chosen as

$$S1_j = S2_j = S3_j = \frac{1}{c}\frac{1-e^{-2cx}}{1+e^{-2cx}}. \tag{41}$$

Then, the following experiments with parameters in Table 1 are performed.

4.1 Training experiments for motion-start region

To enhance the understanding of the friction phenomena in motion-start region, we need to design a set of velocity command to train the weightings of neural network. The command is designed as

$$y = a(1-e^{-mt}), \tag{42}$$

where a is between 25 to 40 *mm/s*, m is between 2 to 9 sec^{-1}. One typical training command is shown in Fig. 7 with a=40 *mm/s*, m=4 sec^{-1}. As illustrated in Fig. 8, the weighting updating stops when the cost function E is smaller than 0.01 $(mm/s)^2$ over 1600 epoches of the step velocity command.

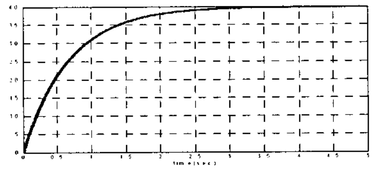

Fig. 7 One typical training command in motion start region

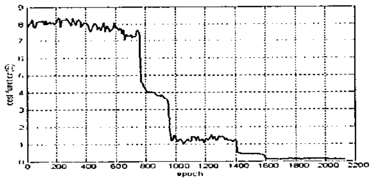

Fig. 8 Cost function vs. epoch during learning process in motion start region

4.2 Training experiments for motion-reverse region

We design a set of velocity command to help us look into the friction phenomena in the motion-reverse region. The sinusoid signals, with frequency form 0.5 *Hz* to 5 *Hz* and amplitude from 30 *mm/s* to 40 *mm/s*, will be the suitable velocity commands because they cover the bandwidth of the system and can fully capture the reverse motion. Fig. 9 describes one typical sinusoidal command with 40 *mm/s* amplitude. Fig. 10 shows the experiment results of cost function. The weighting updating stops when the cost function E is smaller than 0.02 $(mm/s)^2$ over 1400 epoches of the velocity command.

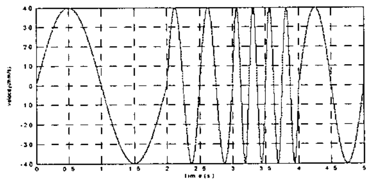

Fig. 9 One typical training command in motion reverse region

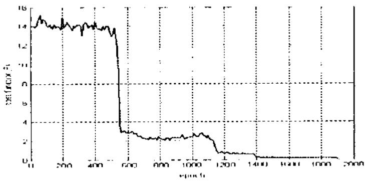

Fig. 10 Cost function vs. epoch during learning process in motion reverse region

4.3 Experiments for a complete sinusoidal command

A sinusoidal velocity command $40\sin(\pi t)$ *mm/s* is fed to the servo loop to verify the compensation performance based on the combined estimated friction from above trained neural network for motion start region and reverse region. The encoder resolution used in this case is $0.0791(\mu m)$. The sample rate for the controller is 2 *kHz*, and velocity controller bandwidth is designed as $\omega_n = 5Hz$. Additionally, the switching times in (39) are set as $t_1 = 0.4\sec$ and $t_2 = 0.6\sec$. The performance comparisons between the proposed method and without friction compensation are presented by two performance indices. The first index is the maximum tracking error at three regions, the motion start region and two velocity reverse regions, is defined as

$$E_{\max} = \max_t |x_r - x| \tag{43}$$

The second index indicating the root mean square tracking error at these three regions is defined as

$$E_{rms} = \sqrt{\frac{1}{N}\sum_N (x_r - x)^2}. \tag{44}$$

Fig. 11 shows the experimental results. In Fig. 11(a), the command and responses of two control systems (with compensation and without compensation) coincide due to the scale used in the ordinate. Fig. 11(b)~Fig. 11(d) illustrate the detail responses at motion start region and motion reverse region. The maximum tracking error and the root mean square tracking error for this case are shown in Table 2 and Table 3, respectively. In motion start region, the maximum tracking error with friction compensation is reduced over 34% and the mean square error is reduced over 53%. In both velocity reverse regions, the two indices are reduced over 48% and 79%, respectively for the first reversal, and over 55% and 90%, respectively for the second reversal. The estimated friction is illustrated in Fig. 12, where the Stribeck effect is captured in the motion start region and then the estimated friction is decreased to Coulomb friction force. Furthermore, there is no Stribeck effect as the motion experiences the velocity reversal region.

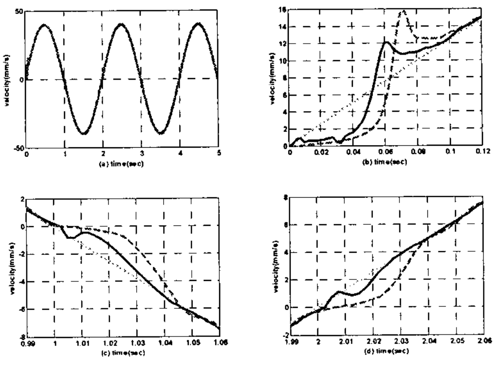

Fig. 11 (a) Velocity command (Dotted Line, $40\sin(\pi t)$ mm/s), velocity response with friction compensation (Solid line), and velocity response without friction compensation (Dashed line) in all trajectory, (b) in motion start region, (c) in the first motion reverse region, and (d) in the second motion reverse region

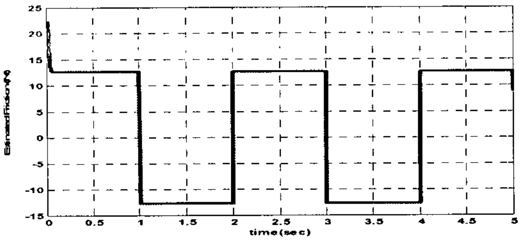

Fig.12 The estimated friction

Table 2 Maximum tracking error index for the velocity command, $40\sin(\pi t)$ *mm/s*

	Without compensation	With Friction compensation	Error Reduction
	$E_{\max\_1}$ (mm/s)	$E_{\max\_2}$ (mm/s)	$\frac{(E_{\max\_1}-E_{\max\_2})}{E_{\max\_1}}\%$
Motion Start	6.98	4.55	34.87
1st Reversal	2.44	1.25	48.67
2nd Reversal	2.07	0.91	55.82

Table 3 Root mean square tracking error index for the velocity command, $40\sin(\pi t)$ *mm/s*

	Without compensation	With Friction compensation	Error Reduction
	E_{rms_1} (mm/s)	E_{rms_2} (mm/s)	$\frac{(E_{rms\_1}-E_{rms\_2})}{E_{rms\_1}}\%$
Motion Start	8.98	4.16	53.71
1st Reversal	1.66	0.35	79.11
2nd Reversal	1.08	0.10	90.90

In addition, two other experimental results are given. One is with a lower frequency velocity command $40\sin(0.4\pi t)$ *mm/s* and the other with a higher frequency command $70\sin(2\pi t)$ *mm/s*. The first velocity command and velocity responses (with and without friction compensation) in the motion start region are shown in Fig. 13. The maximum tracking error and the root mean square tracking error for this case are shown in Table 4, which indicates that the maximum tracking error with friction compensation is reduced over 54% and the mean square error is reduced over 75%. On the other hand, the second velocity command and velocity responses (with and without friction compensation) in the motion reverse region are shown in Fig. 14. The maximum tracking error and the root mean square tracking error for this case are shown in Table 5, which indicates that the maximum tracking error with friction compensation is reduced over 65% and the mean square error is reduced over 88%.

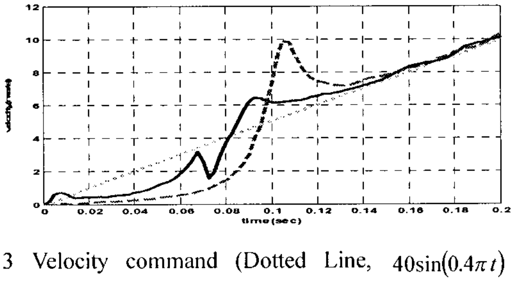

Fig. 13 Velocity command (Dotted Line, $40\sin(0.4\pi t)$ *mm/s*), velocity response with friction compensation (Solid line), and velocity response without friction compensation (Dashed line) in motion start region

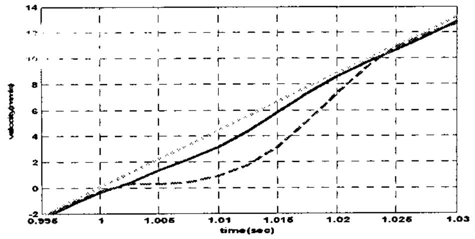

Fig. 14 Velocity command (Dotted Line, $70\sin(2\pi t)$ mm/s), velocity response with friction compensation (Solid line), and velocity response without friction compensation (Dashed line) in motion reverse region

Table 4 Two indices for the velocity command, $40\sin(0.4\pi t)$ *mm/s*, in the motion start region

	Without compensation	With Friction compensation	Error Reduction
	E_1	E_2	$(E_1-E_2)/E_1$ %
$E_{\max}(mm/s)$	4.61	2.09	54.64
$E_{rms}(mm/s)$	2.86	0.69	75.54

Table 5 Two indices for the velocity command, $70\sin(2\pi t)$ *mm/s*, in the motion start region

	Without compensation	With Friction compensation	Error Reduction
	E_1	E_2	$(E_1-E_2)/E_1$ %
$E_{\max}(mm/s)$	3.82	1.31	65.75
$E_{rms}(mm/s)$	4.47	0.53	88.05

5. Conclusion

In this paper, a linear-motor-driven motion system is under study because of the advantages of its simple structure and absence of flexible coupling. The proposed LuGre model-based neural network is designed to be utilized in both the motion-start region and the motion-reverse region for estimating and compensating the friction in a linear motor stage. Comparing the performance without friction compensation, we found the compensation performance of the LuGre MBNN gains advantage in both the motion-start region and the motion reverse region. Finally, the proposed compensator is evaluated and compared experimentally with an uncompensated system on a microcomputer controlled linear motor positioning system. The experimental results show that the velocity error occurrences in the motion-start region and the motion-reverse region are drastically improved.

ACKNOWLEDGEMENT

The author would like to thank the National Science Council of the Republic of China for financial support of this manuscript under Contract No. NSC 92-2212-E-027-011.

REFERENCES

1. Armstrong, Helouvry B., Dupont, P. and Canudas, de Wit C., "A survey of models, analysis tools and compensation methods for the control of machines with friction," Automatica, Vol. 30, No. 7, pp.1083–1138, 1994.

2. Berger, E. J., "Friction modeling for dynamic system simulation." ASME J. of Applied Mechanics Reviews. Vol. 55. No. 6. pp. 535-577, 2002.
3. Karnopp, D., "Computer simulation of stick-slip friction in mechanical dynamic systems," ASME J. of Dynamic systems, Measurement and Control, Vol. 107, No. 1, pp. 100-103, 1985.
4. Sepehri, N., Sassini, F., Lawrence, P. D. and Ghasempoor, A., "Simulation and experimental studies of gear backlash and stick-slip friction in hydraulic excavator swing motion," ASME J. of Dynamic Systems, Measurement and Control, Vol. 118, pp. 463-467,1996.
5. Leine, R. I., Campen, van D. H., Kraker, de A. and Steen,van den L., "Stick-slip vibrations induced by alternate friction models," Nonlinear Dynamics, Vol.16, pp. 41-54, 1998.
6. Armstrong, H. B., "Control of machines with friction," Kluwer Academic Publishers, Nowell, MA, 1991.
7. Dahl, P., "Solid friction damping of mechanical vibrations," AIAA Journal, Vol. 14, No. 12, pp. 1675-1682, 1976.
8. Canudas, de Wit C., Olsson, H., Astrom, K. J. and Lischinsky, P., "A new model for control of systems with friction," IEEE Trans. on Automatic Control, Vol. 40, No. 3, pp. 419-425, 1995.
9. Gao, X. Z. and Ovaska, S. J., "Friction compensation in servo motor systems using neural networks," Proc. Of the 1999 IEEE Midnight sun Workshop on Soft computing Methods in Industrial Applications, Kuusama, Finland, pp.146-151, June 1999.
10. Kovacic, Z., Petik, V. and Bogdan, S., "Neural network-based friction and nonlinear load compensator," Proc. of the 1999 IEEE International Symposium on Industrial Electronics, Vol. 1, pp. 157 - 162, July 1999.
11. Yen, Chen Wen V., Liu, T. Z. and Wang, T. Y., "A neural-network-based velocity tracking control method for direction-changing motions," Control Engineering Practices, Vol.5, No.8, pp. 1071-1076, 1997.
12. Gan, C. and Danai, K., "Model-based recurrent neural network for modeling nonlinear dynamic system," IEEE Transactions on Systems, Man and cybernetics-part b: cybernetics, Vol. 30, No. 2, pp. 344-351, April 2000.
13. Shih, Y.T., "The High Precision Control for a Linear-Motor-Driven Motion Stage with Friction Compensation," Doctor Thesis, National Chiao-Tung University, Hsinchu City, Taiwan, 2004.
14. "PLATINUM DDL and SERVOSTAR setup Guide," Document number: M-LN-016-0702, Kollmorgen, a Danaher Motion Company, June 2002.
15. Edward, P.C., "Digital filtering," Houghton Mifflin Co., Chap. 9, 1992.
16. Werbos, P. J., "Backpropagation Through Time: What it dose and How to do it," Proc. Of the IEEE, Vol. 78, No. 10, pp. 1550-1560, October 1990.