Neural Comput & Applic (2009) 18:567–575
DOI 10.1007/s00521-008-0228-9

ORIGINAL ARTICLE

Robust neural network control system design for linear ultrasonic motor

Chih-Min Lin · Chin-Hsu Leng · Chun-Fei Hsu · Chiu-Hsiung Chen

Received: 13 March 2007 / Accepted: 24 November 2008 / Published online: 17 December 2008

Abstract Linear ultrasonic motor (LUSM) has much merit, such as high precision, fast control dynamics and large driving force, etc.; however, the dynamic characteristic of LUSM is nonlinear and the precise dynamic model of LUSM is difficult to obtain. To tackle this problem, this study presents a robust neural network control (RNNC) system for LUSM to track a reference trajectory with L_2 robust tracking performance. The developed RNNC system is composed of a neural network controller and a robust controller. The neural network controller is the principal controller used to mimic an ideal controller and the robust controller is adopted to achieve L_2 robust tracking performance. The developed RNNC system is then applied to control an LUSM. Experimental results show that the developed RNNC system can achieve favorable tracking performance with unknown of LUSM model.

Keywords Linear ultrasonic motor · Neural network · Robust control

C.-M. Lin (✉) · C.-H. Leng
Department of Electrical Engineering, Yuan-Ze University,
Chung-Li, Tao-Yuan 320, Taiwan, ROC
e-mail: cml@saturn.yzu.edu.tw

C.-F. Hsu
Department of Electrical Engineering, Chung-Hua University,
Hsinchu 300, Taiwan, ROC

C.-H. Chen
Department of Computer Sciences and Information
Engineering, China University of Technology,
Hukou Township 303, Taiwan, ROC

1 Introduction

Modern mechanical systems, such as machine tools and automatic inspection machines, often require high-speed high-accuracy linear motions. These linear motions are usually realized using the rotary motors with a mechanical transmission, such as reduction gears and lead screw. These transmission mechanisms not only significantly reduce the linear motion speed and dynamic response, but also introduce the backlash and large friction. To tackle this problem, linear ultrasonic motor (LUSM) has been introduced to apply the linear motion without using any mechanical transmission. LUSM has much merit, such as high precision, fast control dynamics, large driving force, small dimension, high holding force, silence and high resolution, so that it can be used in many different applications [1]. The driving principle of LUSM is based on the ultrasonic vibration force of piezoelectric ceramic elements and mechanical frictional force. Therefore, its mathematical model is complex and the motor parameters are time-varying because of increasing temperature and changes in motor drive operating conditions [2]. For control system design, the conventional control technologies always need a good understanding of the controlled system; therefore, it is very difficult to control LUSM by using the conventional control technologies. To tackle this problem, some design techniques have been adopted for LUSM control [3–5]; however, these design procedures are overly complex or may cause large chattering in the control efforts which will wear the bearing mechanism and excite unmodelled dynamics.

Recently, the neural network control technique has presented an alternative design method for the control of unknown dynamic systems [6–10]. It is well known that neural network is capable of approximating mapping through choosing adequately learning method. Because of this property, many neural network controllers have been developed for the compensation of the nonlinearities and system uncertainties in control system so that the system performance such as the stability, convergence, and robustness can be improved. However, the structures of the neural networks presented in [6–10] are the feedforward neural networks, which are static mapping schemes. For dynamic mapping, some researches have proposed the recurrent neural networks [11–15]. Since the recurrent neural networks have the internal feedback loops, they capture the dynamic response of systems with feedback through delays. It has been shown that for the dynamic function estimation, the recurrent neural network can achieve superior estimation performance than the feedforward neural network [11].

To tackle the problem of unobtainable of the precise dynamic model of LUSM, this study develops a robust neural network control (RNNC) system to achieve trajectory tracking performance based on the neural network approach and L_2 control technique. The developed RNNC scheme does not require the system model of LUSM, meanwhile it is capable of yielding good tracking performance. The developed RNNC system is composed of a neural network controller and a robust controller. The neural network controller using a recurrent neural network is designed to approximate an ideal controller and the robust controller is developed to guarantee the attenuation of the tracking error to a specified level. Finally, a computer-control-based experimental system is developed to verify the effectiveness of the proposed RNNC design method.

2 Modeling of linear ultrasonic motor

The structure of LUSM is a large face of a relatively thin rectangular piezoelectric ceramic device. The driving principles of LUSM are based on the ultrasonic vibration force of piezoelectric ceramic element and mechanical frictional force [1–3]. Figure 1 shows the principal structure of LUSM considered in this study [4]. The stator vibrator is fitted with bending and longitudinal piezoelectric actuators. They are driven by two electrical sources of identical frequency, but with a phase difference that is carefully controlled. At the vibration tip, an elliptical motion is thus created, resultant of the elliptical and longitudinal motion. The bending actuators convert a large electrical power to mechanical output and the longitudinal actuator dynamically changes the force along the pre-load direction to adjust the frictional force between the stator and the rotor. Friction is inevitable in LUSM. It is a highly complicated process to attempt to build an explicit mathematical friction model for LUSM because friction plays a dual role: it not only contributes to the nonlinear dynamics (e.g., dead zone) of LUSM, but also serves as the driving force for the moving part. Therefore, the dynamic equation of LUSM is very complicated and the parameters of the elements are not easy to obtain.

For developing the control law, LUSM can be described as a second-order nonlinear dynamic equation by the Newton's Law as [4]

$$[M + m(t)]\ddot{d}(t) = F(\mathbf{d}; t) + G(\mathbf{d}; t)u(t) \tag{1}$$

where M is the mass of the moving table; $m(t)$ is the mass of the payload; $\mathbf{d} = [d(t)\ \dot{d}(t)]^T$ represents the position and velocity of the moving table; $F(\mathbf{d};\ t)$ is the nonlinear dynamic function including friction, ripple force and external disturbance; $G(\mathbf{d};\ t)$ is the gain of LUSM resonant inverter; and $u(t)$ is the input force to LUSM. Rewriting Eq. (1), the dynamic equation of LUSM can be obtained as

$$\begin{aligned}\ddot{d}(t) &= \frac{F(\mathbf{d}; t)}{M + m(t)} + \frac{G(\mathbf{d}; t)}{M + m(t)}u(t) \\ &= F_p(\mathbf{d}; t) + G_p(\mathbf{d}; t)u(t)\end{aligned} \tag{2}$$

where $F_p(\mathbf{d};\ t) = \frac{F(\mathbf{d}; t)}{M+m(t)}$ and $G_p(\mathbf{d};\ t) = \frac{G(\mathbf{d}; t)}{M+m(t)}$.Since the model of LUSM is difficult to obtain, a model-free design method termed as robust neural network control (RNNC) system is developed for LUSM control. The LUSM robust neural network control system is shown in Fig. 2. The control objective of this system is to force the output $d(t)$ to follow a reference trajectory $d_m(t)$. Define the tracking error as

$$e(t) = d_m(t) - d(t) \tag{3}$$

where $d_m(t)$ is the reference trajectory specified by a reference model following a command input $d_c(t)$. If the dynamic function of the nonlinear system in Eq. (2) is well known, there exits an ideal controller as [16]

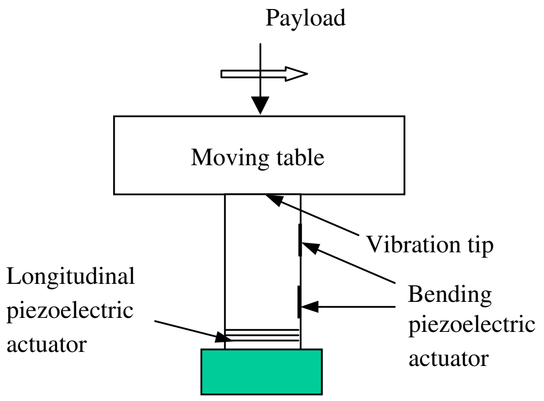

Fig. 1 Functional structure of the piezoelectric-type linear ultrasonic motor

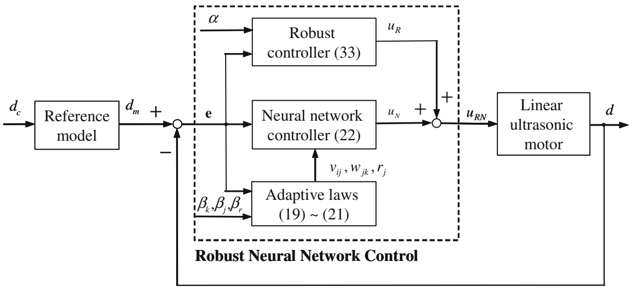

Fig. 2 LUSM robust neural network control system

$$u^*(t) = G_p(\mathbf{d}; t)^{-1}[-F_p(\mathbf{d}; t) + \ddot{d}_m(t) + k_2\dot{e}(t) + k_1 e(t)] \tag{4}$$

Applying Eq. (4) to Eq. (2), gives the following tracking error equation

$$\ddot{e}(t) + k_2\dot{e}(t) + k_1 e(t) = 0 \tag{5}$$

If k_1 and k_2 are chosen to correspond to the coefficients of a Hurwitz polynomial, that is a polynomial whose roots lie strictly in the open left half of the complex plane, then $\lim_{t\to\infty} e(t) = 0$. Due to the nonlinearity and time-varying characteristics of LUSM, the precise dynamic functions of F_p(d; t) and G_p(d; t) are unobtainable in practical applications. Therefore, the ideal controller $u^*(t)$ can not be implemented. Thus, a robust neural network control system is proposed to control LUSM.

3 Structure of recurrent neural network

In this study, a recurrent neural network will be utilized as the principal controller of the LUSM control system. A three-layer recurrent neural network is shown in Fig. 3, which consists of an input layer (the i layer), a hidden layer (the j layer) and an output layer (the k layer). The signal propagation and the basic function in each layer are shown as follows:

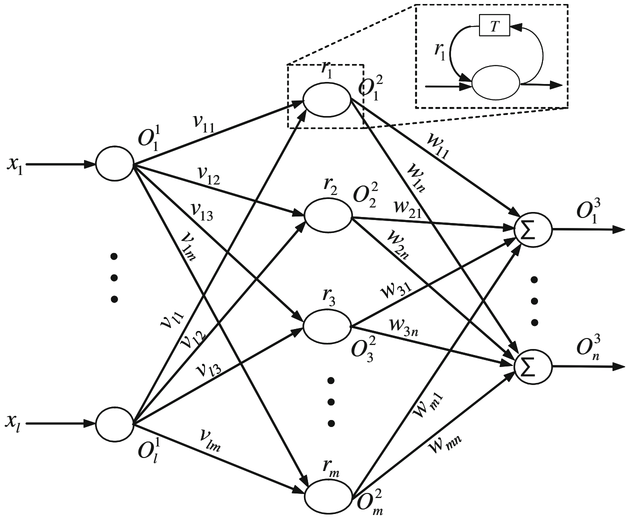

Fig. 3 Structure of the recurrent neural network

3.1 Layer 1: input layer

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

$$\mathrm{net}_i^1 = x_i \tag{6}$$

$$O_i^1 = f_i^1(\mathrm{net}_i^1) = \mathrm{net}_i^1 \tag{7}$$

where x_i, $i = 1, 2,\ldots, l$ are the input variables; and $f_i^1(\cdot)$ is the activation function of the input layer, which is chosen as an unity function in this layer.

3.2 Layer 2: hidden layer

In this layer, each node performs an output represented as

$$\mathrm{net}_j^2 = r_j O_{jT}^2 + \sum_{i=1}^{l} v_{ij} O_i^1 \tag{8}$$

$$O_j^2 = f_j^2(\mathrm{net}_j^2) = \frac{1}{1 + e^{-\mathrm{net}_j^2}} \tag{9}$$

where v_{ij}, $j = 1, 2,\ldots, m$ are the connect weights between the input and the hidden layers; r_j is the internal feedback gain of O_j^2; O_{jT}^2 denotes the output value of O_j^2 through the delay time T; and $f_j^2(\cdot)$ is the activation function of the hidden layer, which is chosen as a sigmoid function in this layer. Since the network contains a delayed feedback, it possesses the dynamic characteristic.

3.3 Layer 3: output layer

The node k in this layer is labeled as Σ, which computes the overall output as the summation of all incoming signals

$$\mathrm{net}_k^3 = \sum_{j=1}^{m} w_{jk} O_j^2 \tag{10}$$

$$O_k^3 = f_k^3(\mathrm{net}_k^3) = \mathrm{net}_k^3 \tag{11}$$

where w_{jk}, $k = 1, 2,\ldots, n$ are the connect weights between the hidden and the output layers; and $f_k^3(\cdot)$ is the activation function of the output layer, which is chosen as an unity function in this layer. O_k^3 is the kth output variable.

4 Design of robust neural network controller

The robust neural network controller is composed of a neural network controller and a robust controller

$$u_{\mathrm{RN}}(t) = u_N(t) + u_R(t) \tag{12}$$

where the neural network controller $u_N(t)$ using a recurrent neural network is designed to approximate the ideal controller and the robust controller $u_R(t)$ is adopted to achieve specified robust tracking performance. By substituting Eq. (12) into Eq. (2) and combining with Eq. (4), the tracking error dynamic equation can be obtained as follows:

$$\dot{\mathbf{e}} = \mathbf{A}_{\mathrm{m}}\mathbf{e} + G_p(\mathbf{d};t)\mathbf{b}_{\mathbf{m}}(u^*(t) - u_N(t) - u_R(t)) \tag{13}$$

where $\mathbf{e} = [e(t)\ \dot{e}(t)]^T$, $\mathbf{A}_{\mathbf{m}} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$ and $\mathbf{b}_{\mathbf{m}} = [0 \quad 1]^T$.

4.1 A. On-line learning algorithm

In the following, the learning algorithm of the recurrent neural network is derived to achieve favorable approximation performance. Define a cost function $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$, then $\dot{V} = \mathbf{e}^T\dot{\mathbf{e}}$. Multiplying both sides by e^T in Eq. (13), yields

$$\mathbf{e}^T\dot{\mathbf{e}} = \mathbf{e}^T\mathbf{A}_{\mathrm{m}}\mathbf{e} + \mathbf{e}^T G_p(\mathbf{d};t)\mathbf{b}_{\mathrm{m}}(u(t)^* - u_N(t) - u_R(t)) \tag{14}$$

In order to train the recurrent neural network, the online learning algorithm of the recurrent neural network controller is a gradient descent search algorithm in the space of network parameters, and aims to minimize the function $\dot{V} = \mathbf{e}^T\dot{\mathbf{e}}$. According to the chain rule of the gradient descent method, the weights in the output layer are updated by the following equation

$$\dot{w}_{jk} = -\eta_k \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial w_{jk}} = -\eta_k \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial \mathrm{net}_k^3}\frac{\partial \mathrm{net}_k^3}{\partial w_{jk}} = -\eta_k \delta_k O_j^2 \tag{15}$$

where the approximation error term needs to be calculated and back-propagated by

$$\delta_k \overset{\Delta}{=} \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial \mathrm{net}_k^3} = \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial u_N}\frac{\partial u_N}{\partial O_k^3}\frac{\partial O_k^3}{\partial \mathrm{net}_k^3} = -e^T G_p(\mathbf{d};t)\mathbf{b}_{\mathrm{m}} \tag{16}$$

The weights in the hidden layer are updated by

$$\begin{aligned}\dot{v}_{ij} &= -\eta_j \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial v_{ij}} = -\eta_j \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial \mathrm{net}_k^3}\frac{\partial \mathrm{net}_k^3}{\partial O_j^2}\frac{\partial O_j^2}{\partial \mathrm{net}_j^2}\frac{\partial \mathrm{net}_j^2}{\partial v_{ij}} \\ &= -\eta_j f_j^{2'}(\mathrm{net}_j^2)O_i^1 \sum_k w_{jk}\delta_k\end{aligned} \tag{17}$$

where $f_j^{2'}(net_j^2)$ is the derivative of activation function. The weights in the recurrent signal are updated by

$$\begin{aligned}\dot{r}_j &= -\eta_r \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial r_j} = -\eta_r \frac{\partial \mathbf{e}^T\dot{\mathbf{e}}}{\partial \mathrm{net}_k^3}\frac{\partial \mathrm{net}_k^3}{\partial O_j^2}\frac{\partial O_j^2}{\partial \mathrm{net}_j^2}\frac{\partial \mathrm{net}_j^2}{\partial r_j} \\ &= -\eta_r f_j^{2'}(\mathrm{net}_j^2)O_{jT}^2 \sum_k w_{jk}\delta_k\end{aligned} \tag{18}$$

where η_k, η_j and η_r are the learning-rates with positive constant. According to the unavailable system dynamics, G_p(d; t) in Eq. (16) is rewritten as $|G_p(\mathrm{d};\ t)|\mathrm{sgn}(G_p(\mathrm{d};\ t))$. Therefore, the update laws of the neural network controller shown in Eqs. (15), (17) and (18) can be rewritten as follows:

$$\dot{w}_{jk} = -\beta_k \tau_k O_j^2 \tag{19}$$

$$\dot{v}_{ij} = -\beta_j f_j^{2'}(\mathrm{net}_j^2)O_i^2 \sum_k w_{jk}\tau_k \tag{20}$$

$$\dot{r}_j = -\beta_r f_j^{2'}(\mathrm{net}_j^2)O_{jT}^2 \sum_k w_{jk}\tau_k \tag{21}$$

where $\tau_k = -\mathrm{sgn}[G_p(\mathbf{d};\ t)]\mathbf{e}^T\mathbf{b}_{\mathbf{m}}$, the terms β_k, β_j and β_r are some positive constants absorbing the control gain $|G_p(\mathbf{d};\ t)|$ and representing the new learning-rates. Consequently, only the sign of G_p(d; t) is required in the design procedure, and it can be easily obtained from the physical characteristic of the controlled system.

4.2 B. Robust controller design with L_2 tracking performance

In the following, a robust controller will be developed to achieve L_2 robust tracking performance for the approximation error between the neural network controller and the ideal controller. Suppose there exists a neural network controller to approximate the ideal controller such that

$$u^*(t) = u_N(\hat{\Theta}, t) + \varepsilon(t) \tag{22}$$

where $\hat{\Theta} = [\hat{w}_{jk}\ \hat{v}_{ij}\ \hat{r}_j]^T$ is the estimated weighting vector of the neural network controller; and ε denotes the matching error between the neural network controller u_N and the ideal controller u^*. While ε appears, the following L_2 tracking performance is defined [17]

$$\int_0^T \mathbf{e}^T(t)\mathbf{Q}\mathbf{e}(t)\,\mathrm{d}t \le \mathbf{e}^T(0)\mathbf{P}\mathbf{e}(0) + \delta^2 \int_0^T \varepsilon^2(t)\,\mathrm{d}t, \qquad \forall T \in [0,\infty), \varepsilon \in L_2[0,T] \tag{23}$$

for given constant weighting matrices $\mathbf{Q} = \mathbf{Q}^T \ge 0$ and $\mathbf{P} = \mathbf{P}^T \ge 0$, and a prescribed attenuation constant δ. If the system starts with initial condition $\mathbf{e}(0) = 0$, then the L_2 tracking performance in Eq. (23) can be rewritten as

$$\sup_{\varepsilon \in L_2[0,T]} \frac{\|\mathbf{e}\|_{\mathrm{Q}}}{\|\varepsilon\|_2} \le \delta \tag{24}$$

where $\|\mathbf{e}\|_{\mathbf{Q}}^2 = \int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\ \mathrm{d}t$ and $\|\varepsilon\|_2^2 = \int_0^T \varepsilon^2\ \mathrm{d}t$, i.e., the L_2-gain from ε to the tracking error $\mathbf{e}$ must be equal to or less than δ.

By substituting Eq. (22) into Eq. (13), the tracking error dynamic equation can be rewritten as

$$\dot{\mathbf{e}} = \mathbf{A}_{\mathbf{m}}\mathbf{e} + G_p(\mathbf{d};t)\mathbf{b}_{\mathbf{m}}(\varepsilon - u_R) \tag{25}$$

Then the following theorem can be stated and proved.

Theorem 1: Consider a linear ultrasonic motor presented by Eq. (2). If the robust neural network control system is designed as Eq. (12), where the update laws of the neural network controller are designed as Eqs. (19)–(21) and the robust controller is designed as

$$u_R = \frac{1}{\kappa G_p(\mathbf{d};t)}\mathbf{b}_{\mathrm{m}}^T\mathbf{Pe} \tag{26}$$

where κ is a positive factor and $\mathbf{P} = \mathbf{P}^T$ is a solution of the following Riccati-like equation

$$\mathbf{PA}_{\mathrm{m}} + \mathbf{A}_{\mathrm{m}}^T\mathbf{P} + \mathbf{Q} - \frac{2}{\kappa}\mathbf{Pb}_{\mathrm{m}}\mathbf{b}_{\mathrm{m}}^T\mathbf{P} + \frac{1}{\rho^2}\mathbf{Pb}_{\mathrm{m}}\mathbf{b}_{\mathrm{m}}^T\mathbf{P} = 0 \tag{27}$$

where ρ is a specified positive constant, then the L_2 tracking performance in Eq. (23) can be achieved.

Proof: Define a Lyapunov function in the following form

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} \tag{28}$$

Taking the derivative of the Lyapunov function and using Eqs. (25) and (26), yields

$$\begin{aligned}\dot{V} &= \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}}\\ &= \frac{1}{2}[\mathbf{A}_{\mathrm{m}}\mathbf{e} + G_p(\mathbf{d};t)\mathbf{b}_{\mathrm{m}}(\varepsilon - u_R)]^T\mathbf{Pe}\\ &\quad + \frac{1}{2}\mathbf{e}^T\mathbf{P}[\mathbf{A}_{\mathrm{m}}\mathbf{e} + G_p(\mathbf{d};t)\mathbf{b}_{\mathrm{m}}(\varepsilon - u_R)]\\ &= \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{\mathrm{m}}^T\mathbf{P} + \mathbf{PA}_{\mathrm{m}})\mathbf{e} + \mathbf{e}^T\mathbf{P}G_p(\mathbf{d};t)\\ &\quad \times \mathbf{b}_{\mathrm{m}}(\varepsilon - \frac{1}{\kappa G_p(\mathbf{d};t)}\mathbf{b}_{\mathrm{m}}^T\mathbf{Pe})\end{aligned} \tag{29}$$

By straightforward manipulations from Riccati-like equation (27), Eq. 29) can be rewritten as

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{e}^T(\mathbf{Q} + \frac{1}{\rho^2}\mathbf{Pb}_{\mathrm{m}}\mathbf{b}_{\mathrm{m}}^T\mathbf{P})\mathbf{e} + \frac{1}{2}\varepsilon\mathbf{b}_{\mathrm{m}}^T G_p(\mathbf{d};t)\mathbf{Pe}\\ &\quad + \frac{1}{2}\mathbf{e}^T\mathbf{P}G_p(\mathbf{d};t)\mathbf{b}_{\mathrm{m}}\varepsilon\\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \frac{1}{2}(\frac{1}{\rho}\mathbf{b}_{\mathrm{m}}^T\mathbf{Pe} - G_p(\mathbf{d};t)\rho\varepsilon)^T\\ &\quad \times (\frac{1}{\rho}\mathbf{b}_{\mathrm{m}}^T\mathbf{Pe} - G_p(\mathbf{d};t)\rho\varepsilon) + \frac{1}{2}G_p^2(\mathbf{d};t)\rho^2\varepsilon^2\\ &\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \frac{1}{2}G_p^2(\mathbf{d};t)\rho^2\varepsilon^2\\ &\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \frac{1}{2}\delta^2\varepsilon^2\end{aligned} \tag{30}$$

where $\varepsilon\mathbf{b}_{\mathbf{m}}^T G_p(\mathbf{d};t)\mathbf{Pe} = \mathbf{e}^T\mathbf{P}G_p(\mathbf{d};t)\mathbf{b}_{\mathbf{m}}\varepsilon$ is used since it is a scale, and the new prescribed attenuation constant is defined as $\rho|G_p(\mathbf{d};t)| \le \delta$. Integrating the above equation from $t = 0$ to $t = T$ yields

$$V(T) - V(0) \le -\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Qe}\,\mathrm{d}t + \frac{1}{2}\delta^2\int_0^T \varepsilon^2\,\mathrm{d}t \tag{31}$$

Since $V(T) \ge 0$, the above inequality implies the following inequality

$$\begin{aligned}\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Qe}\,\mathrm{d}t &\le V(0) + \frac{1}{2}\delta^2\int_0^T \varepsilon^2\,\mathrm{d}t\\ &= \frac{1}{2}\mathbf{e}^T(0)\mathbf{Pe}(0) + \frac{1}{2}\delta^2\int_0^T \varepsilon^2\,\mathrm{d}t\end{aligned} \tag{32}$$

As the result, the robust neural network control system achieves the L_2 robust tracking performance in the Lyapunov sense.

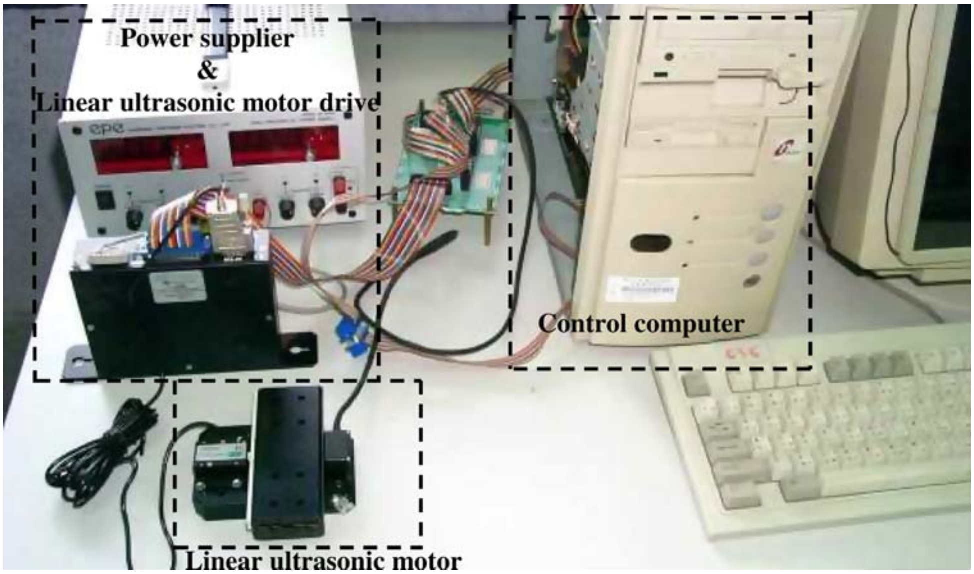

Fig. 4 Linear ultrasonic motor computer control system

In the implementation of the robust controller, since the control gain $G_p(\mathbf{d};\ t)$ may be unknown, Eq. (26) can be rewritten as

$$u_R = \frac{1}{\alpha \operatorname{sgn}(G_p(\mathbf{d};t))}\mathbf{b}_{\mathrm{m}}^T\mathbf{Pe} \tag{33}$$

where $\alpha = \boldsymbol{\kappa}|G_p(\mathbf{d};\ t)|$ is a positive constant absorbing the control gain $|G_p(\mathbf{d};\ t)|$.

5 Experimental results

The computer control experimental system for LUSM is shown in Fig. 4. A servo control card is installed in the control computer, which included multi-channels of D/A, A/D, PIO and encoder interface circuits. The position of the moving table is fed back using a linear scale. The proposed robust neural network control system is realized in the Pentium using the "Turbo C" language. The control interval of the control system is set as 2 ms. Two test conditions are provided in the experiments, which are the nominal condition and the payload condition. The payload condition is the addition of one iron disk with 4.3 kg weight to the mass of the moving table which is with 0.9 kg weight. The control objective is to control the moving table to follow a 0.02 m periodic step command. Moreover, a second-order transfer function $\frac{64}{s^2+16s+64}$ is chosen as the reference model for the step command.

The neural network is chosen with 2 input, 7 hidden-layer neurons, and 1 output; thus, there are 28 adjustable weights. In Eq. (19) the parameter β_k is the leaning-rate of the interconnection weight, w_{jk}, between the hidden and output layers. In Eq. (20) the parameter β_j is the leaning-rate of the interconnection weight, v_{ij}, between the input and hidden layers. In Eq. (21) the parameter β_r is the leaning-rate of the interconnection weight, r_j, of hidden layer. If these learning-rates are chosen too small, then the parameter convergence of the RNNC system will be easily achieved; however, this will result in slow learning speed. On the other hand, if the learning-rates are chosen too large, then the learning speed will be fast; however, the RNNC system may become more unstable for the parameter convergence. δ in Eq. (24) represents the attenuation gain of the tracking error. Solving the Riccati-like equation (27) with $\kappa = 2\rho^2$, it is obtained that

$$\mathbf{Q} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} 1.7625 & 0.7812 \\ 0.7812 & 0.8088 \end{bmatrix} \tag{34}$$

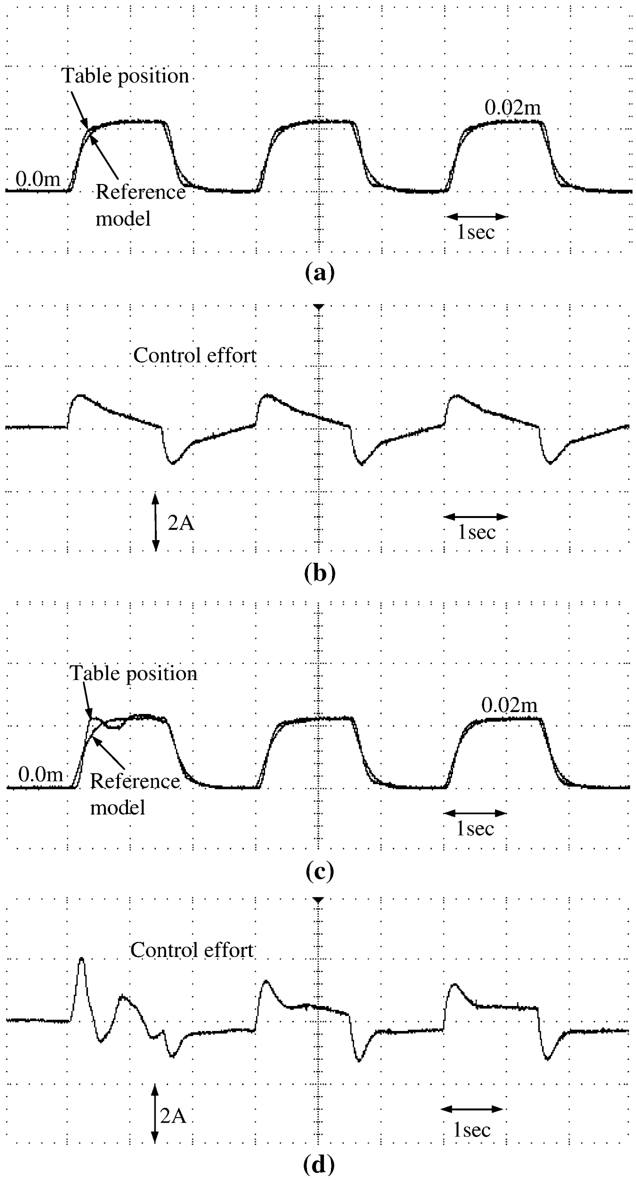

Fig. 5 Experimental results of LUSM control system with $\beta_k = \beta_j = \beta_r = 10$ and $\delta = 0.45$

First, consider the small learning-rates with $\beta_k = \beta_j = \beta_r = 10$ and attenuation gain $\delta = 0.45$. With these parameters, the experimental results of LUSM control system for two test conditions are shown in Fig. 5. The tracking responses are shown in Fig. 5a and c; and the associated control efforts are shown in Fig. 5b and d for nominal condition and payload condition, respectively. Simulation results show that tracking performance is not good enough because the learning-rates β_k, β_j and β_r are too small and the attenuation gain δ is too large. To speed up the convergence, the learning-rates are increased as $\beta_k = \beta_j = \beta_r = 50$ and δ is retained as $\delta = 0.45$. With these parameters, the experimental results of LUSM control system for two test conditions are shown in Fig. 6. The tracking responses are shown in Fig. 6a and c; and the associated control efforts are shown in Fig. 6b and d for nominal condition and payload condition, respectively. Comparing to Fig. 5, it is shown that Fig. 6 can achieve better tracking performance by choosing larger

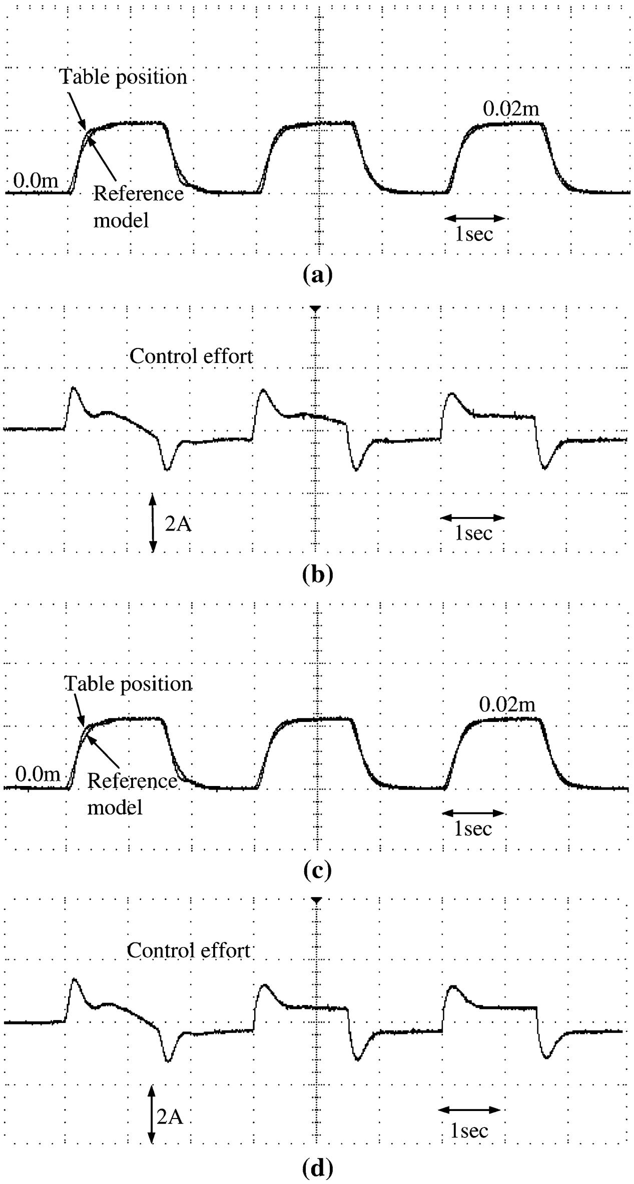

Fig. 6 Experimental results of LUSM control system with $\beta_k = \beta_j = \beta_r = 50$ and $\delta = 0.45$

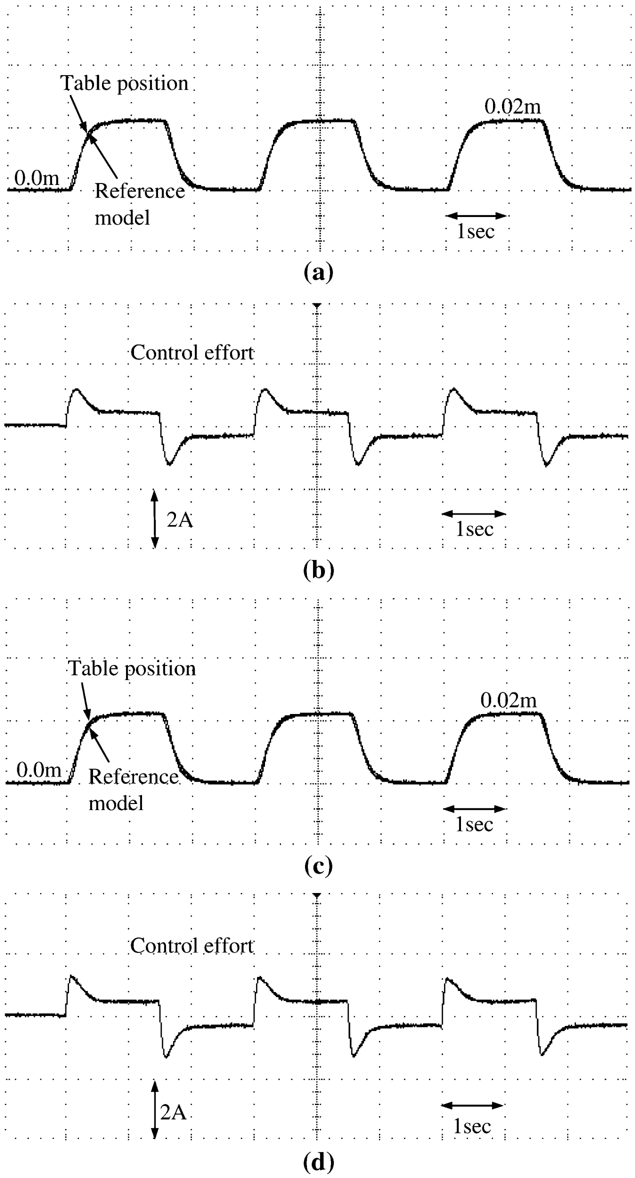

Fig. 7 Experimental results of LUSM control system with $\beta_k = \beta_j = \beta_r = 50$ and. $\delta = 0.225$

learning-rates. Moreover, by specifying $\delta = 0.225$ to increase L^2 tracking performance and the learning-rates are retained as $\beta_k = \beta_j = \beta_r = 50$, the experimental results for two test conditions are shown in Fig. 7. The tracking responses are shown in Fig. 7a and c; and the associated control efforts are shown in Fig. 7b and d for nominal condition and payload condition, respectively. From the experimental results, it can be seen that robust tracking performance can be further improved as the attenuation gain δ is decreased. For observing the on-line learning processes, with $\beta_k = \beta_j = \beta_r = 50$ and $\delta = 0.225$, the time responses of neural network weighs w_{jk}, v_{ij} and r_j for nominal condition and payload condition are shown in Figs. 8 and 9, respectively. The responses of weight w_{j1} are shown in Figs. 8a and 9a; the responses of weight v_{1j} are shown in Figs. 8b and 9b; the responses of weight v_{2j} are shown in Figs. 8c and 9c; and the responses of weight r_j are shown in Figs. 8d and 9d, respectively. These results show that the neural network weights can be on-line tuned to achieve favorable tracking performance.

6 Conclusions

This study has successfully demonstrated the design and implementation of a robust neural network control (RNNC) system for the position control of a linear ultrasonic

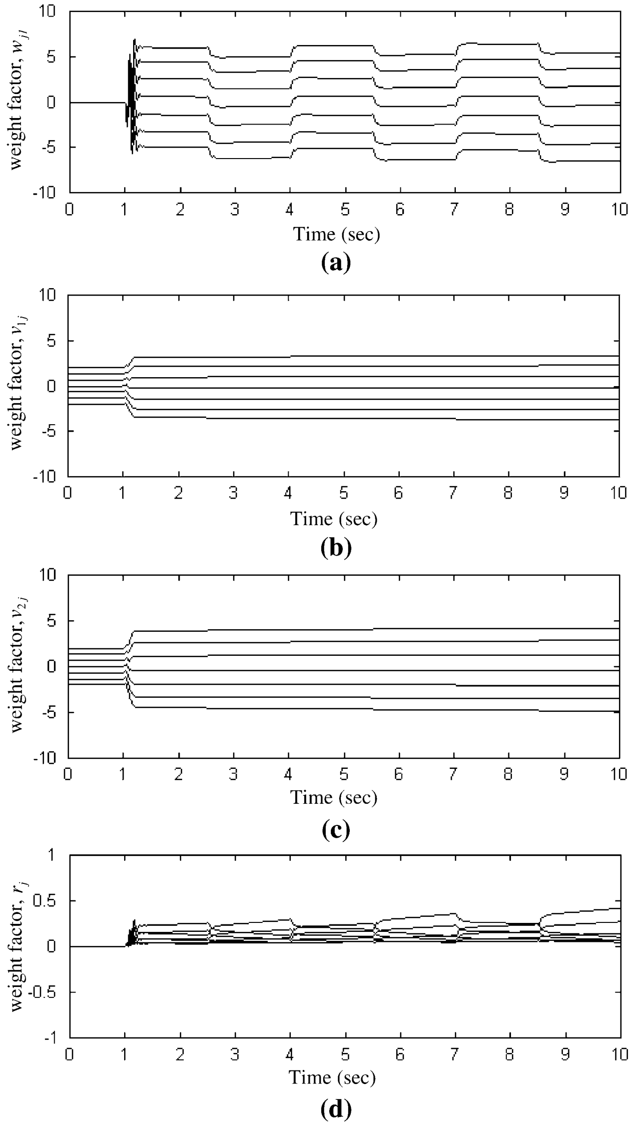

Fig. 8 Time responses of neural network weights for nominal condition

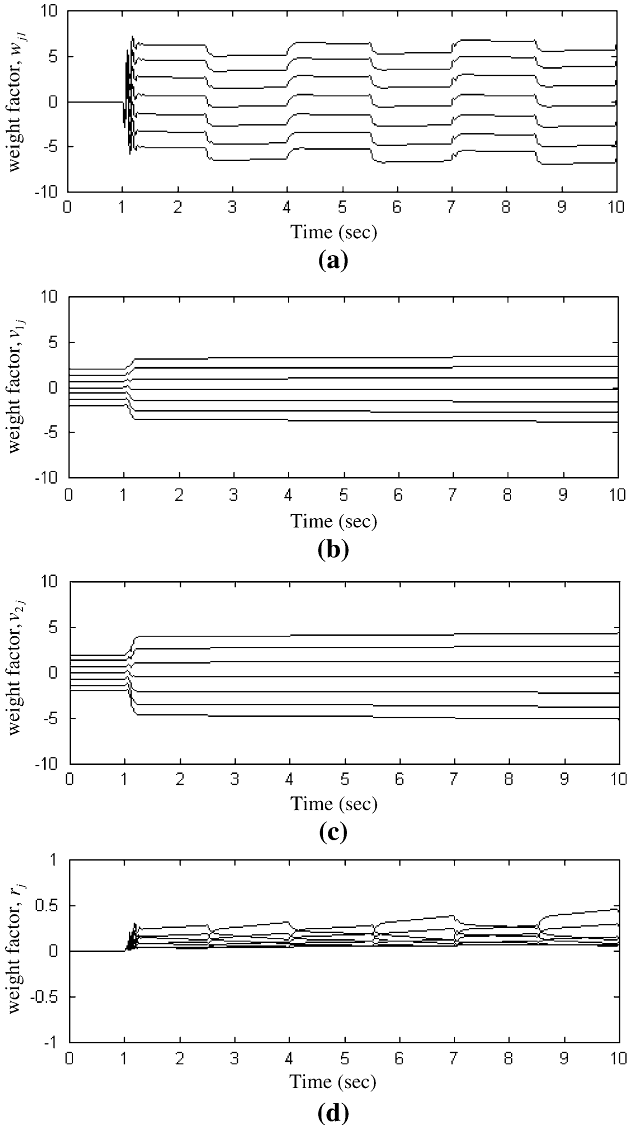

Fig. 9 Time responses of neural network weights for payload condition

motor (LUSM) system. Since the dynamic characteristic of LUSM is nonlinear and the precise dynamic model is unobtainable, the developed model-free RNNC system has been applied to achieve precision position control of LUSM. Experimental results have demonstrated the effectiveness of the proposed design method. The major contributions of this paper are: (1) the successful development of a RNNC system, in which the gradient descent method is used to derive the on-line learning algorithm. (2) the L_2 tracking performance can be achieved with a desired attenuation level. (3) the successful applications of RNNC system to control LUSM. (4) the proposed model-free control methodology can be easily extended to other motors. In this study, the experimental system is implemented on PC-based, future work on the proposed RNNC design includes the hardware implementation using the high-performance microprocessor or digital signal processor (DSP) to provide flexible environments with high execution rate. As shown in the experiments, different learning-rates of neural network will affect the control performance. Another future study could be the derivation of optimal learning-rate of neural network for achieving better control performance. In order to further improve control performance, the other future study can be the derivation of mathematical model of LUSM and the controller design with partially known of system dynamics.

Acknowledgments The authors appreciate the financial support in part from National Science Council of the Republic of China under Grant NSC93-2213-E-155-038. The authors would like to express their gratitude to the Reviewers for their valuable comments.

References

1. Sashida T, Kenjo T (1993) An introduction to ultrasonic motors. Clarendon Press, Oxford
2. He S, Chen W, Tao X, Chen Z (1998) Standing wave bi-directional linearly moving ultrasonic motor. IEEE Trans Ultrason Ferroelectr Freq Control 45(5):1133–1139. doi:10.1109/58.726435
3. Tan KK, Lee TH, Zhou HX (2001) Micro-positioning of linear-piezoelectric motors based on a learning nonlinear PID controller. IEEE/ASME Trans Mechatoron 6(4):428–436
4. Wai RJ, Lin CM, Peng YF (2004) Adaptive hybrid control for linear piezoelectric ceramic motor drive using diagonal recurrent CMAC network. IEEE Trans Neural Netw 15(6):1491–1506. doi:10.1109/TNN.2004.837784
5. Lin CM, Peng YF (2004) Adaptive CMAC-based supervisory control for uncertain nonlinear systems. IEEE Trans Syst Man Cybern B 34(2):1248–1260. doi:10.1109/TSMCB.2003.822281
6. Xinghuo Y, Efe MO, Kaynak O (2002) A general backpropagation algorithm for feedforward neural networks learning. IEEE Trans Neural Netw 13(1):251–254. doi:10.1109/72.977323
7. Lin CM, Hsu CF (2002) Neural-network-based adaptive control for induction servomotor drive system. IEEE Trans Ind Electron 49(1):115–123. doi:10.1109/41.982255
8. Duarte-Mermoud MA, Suarez AM, Bassi DF (2005) Multivariable predictive control of a pressurized tank using neural networks. Neural Comput Appl 15(1):18–25. doi:10.1007/s00521-005-0003-0
9. Yu DL, Yu DW (2007) A new structure adaptation algorithm for RBF networks and its application. Neural Comput Appl 16(1):91–100. doi:10.1007/s00521-006-0067-5
10. Hsu CF (2007) Self-organizing adaptive fuzzy neural control for a class of nonlinear systems. IEEE Trans Neural Netw 18(4):1232–1241. doi:10.1109/TNN.2007.899178
11. Lee CH, Teng CC (2000) Indentification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Trans Fuzzy Syst 8(4):349–366. doi:10.1109/91.868943
12. Lin CM, Hsu CF (2002) Recurrent neural network adaptive control of wing rock motion. J Guid Dyn Contr 25(6):1163–1165. doi:10.2514/2.4998
13. Mastorocostas PA, Theocharis JB (2002) A recurrent fuzzy-neural model for dynamic system identification. IEEE Trans Syst Man Cybern B 32(2):176–190. doi:10.1109/3477.990874
14. Lin CM, Hsu CF (2003) Neural network hybrid control for antilock braking systems. IEEE Trans Neural Netw 14(2):351–359. doi:10.1109/TNN.2002.806950
15. Lin CM, Hsu CF (2004) Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings. IEEE Trans Fuzzy Syst 12(5):733–742. doi:10.1109/TFUZZ.2004.834803
16. Slotine JJE, Li WP (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs
17. Chen BS, Lee CH (1996) H^{∞} tracking design of uncertain nonlinear SISO systems: adaptive fuzzy approach. IEEE Trans Fuzzy Syst 4(1):32–43. doi:10.1109/91.481843