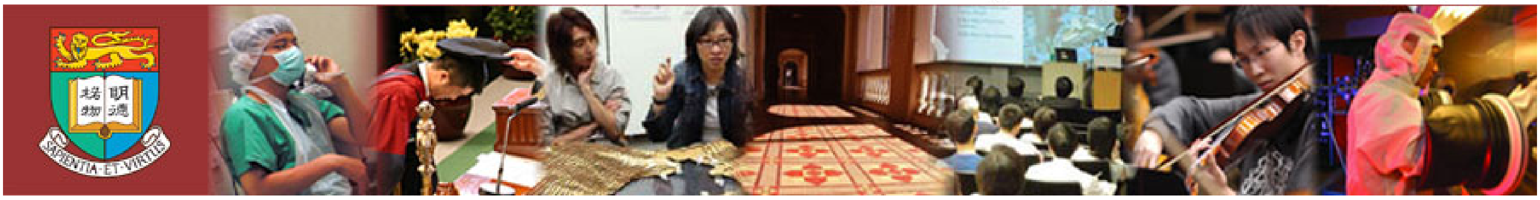


Title	Neuro-fuzzy speed tracking control of traveling-wave ultrasonic motor drives using direct pulse width modulation
Author(s)	Chau, KT; Chung, SW
Citation	Conference Record - Ias Annual Meeting (Ieee Industry Applications Society), 2002, v. 3, p. 2043-2050
Issued Date	2002
URL	http://hdl.handle.net/10722/46350
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Neuro-Fuzzy Speed Tracking Control of Traveling-Wave Ultrasonic Motor Drives Using Direct Pulse Width Modulation

K. T. Chau and S. W. Chung
The University of Hong Kong
Pokfulam Road, Hong Kong, China

Abstract—**The traveling-wave ultrasonic motor (TUSM) drive offers many distinct advantages but suffers from severe system nonlinearities and parameter variations especially during speed control. This paper presents a new speed tracking control system for the TUSM drive, which newly incorporates neuro-fuzzy control and direct pulse width modulation to solve the problem of nonlinearities and variations. Increasingly, the proposed control system is digitally implemented by a low-cost digital signal processor (DSP) based microcontroller, hence reducing the system hardware size and cost. Experimental results confirm that the proposed speed tracking controller can offer good steady-state and transient performances.**

Keywords—ultrasonic motor; neuro-fuzzy; control; drive; pulse width modulation

I. INTRODUCTION

Recently, the traveling-wave ultrasonic motor (TUSM) drive has become attractive for servo applications because of its advantages of high torque at low speed, lightweight, compact size, fast response, no electromagnetic interference (EMI), and quiet operation. Hence, TUSMs are very suitable in mechatronic applications, especially direct-servo, or gearless drive, applications. However, the TUSM drive exhibits heavily nonlinear behavior, and high sensitivity to temperature change. The drift of temperature-dependent parameters is particularly significant during the speed tracking control of prolonged operation. Such peculiarities present quite a challenge for servo speed control using TUSMs.

Basically, the speed of the TUSM can be manipulated by controlling the frequency, phase difference and the voltage amplitude of the two sinusoidal voltage waveforms which input to the TUSM. Many researchers have opted for the frequency of the sinusoidal voltage waveforms as the control variable [1]-[5]. It can easily be controlled digitally and spans the whole speed range the TUSM allows. Another approach is to use frequency for speed control and boost converters for balancing the voltage amplitudes [6]-[9]. Some of them employ the voltage-controlled-oscillators (VCOs). Because the VCO outputs a narrow range of frequency over a high frequency (40~43kHz), the circuit can easily be compromised by electrical noise or tolerances of the electronic components. Furthermore, boost converters add bulk to the overall system and increase the hardware cost. Alternatively, there have also been reports of using phase difference together with the frequency for servo position control to provide quick response by means of signal function generator [10], [11]. However, using signal function generator or microcontroller to digitally vary the phase difference of the two voltages leads to significant audio noise, especially in the case of speed control, hence losing the advantage of quietness of using TUSM [12]. In certain robotic application, some researchers also employ PWM method to control the turn on and off time of the voltage sources over several ultrasonic cycles [13]. The variation of the duty ratio can lead to a change of the average speed the TUSM attains. However, it also suffers from very audible noise.

Recently, it has been found that utilizing frequency together with PWM of the dc voltage directly supplied to the TUSM can greatly improve the overall control system [12]. Current DSP microcontroller cannot output sufficient resolution for frequency control. By incorporating duty ratio as another control variable, this resolution problem can be solved without resorting to adding extra analog device. As a result, the hardware cost can be reduced. Furthermore, such approach allows all control signals in digital form, thus improving the noise immunity. In addition, the speed of TUSM displays partially linear response to duty ratio. This further improves the control performance.

On the other hand, there are attempts of modeling for TUSM. However, the resulting models are so complex or data demanding that they are impractical for real-time control [14]-[16]. Alternatively, many empirical approaches for speed control, such as classical PI control [7], [12], fuzzy logic control [1], adaptive control [2], [17], neural network control [5] have been proposed. PI controller is simple and offers wide stability margin. However, it requires meticulous fine-tuning and cannot cope with varying driving conditions. Fuzzy logic control can compensate the nonlinear behavior of TUSM through human expertise. Yet it relies too much on the intuition and experience of the designer. It also requires tedious fine tuning. Both adaptive control and neural network adapt the controller to the TUSM by itself, thus suppress the disadvantages of fuzzy logic and PI control. Adaptive control identifies the drifting parameters and adjusts the controller accordingly. Unfortunately, adaptive control requires lengthy computation and explicitly assumes a linear model of the system, which is definitely not the case of TUSM. Neural network consists of layers of nodes. Each node is connected by weighted links. By tuning the weights of the network on-line, it can handle the complicated nonlinear relationships between input and outputs. On the other hand, neural network suffers from the fact that the corresponding input-output relationships are not visible.

There is a trend of employing fuzzy neural network control, or neuro-fuzzy control, to perform motion control [4],

0-7803-7420-7/02/$17.00 © 2002 IEEE

[10], [18], [19], [20]. This control incorporates the fuzzy logic process into a neural network. Excellent performance has been achieved by employing this control strategy for servo position control [10]. There has also been attempt to employing the fuzzy neural network to adjust the gains of the PI speed controller [4]. However, there are difficulties in establishing the fuzzy rules. In case of position control, the rules to describe the should-be behavior of the control input to position error are not obvious. The rules are also not apparent in adjusting the gains of the PI speed controller.

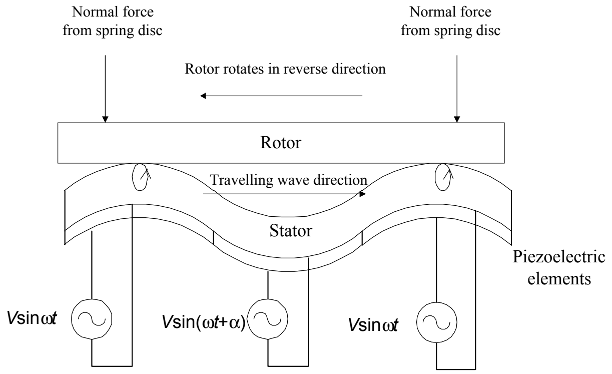
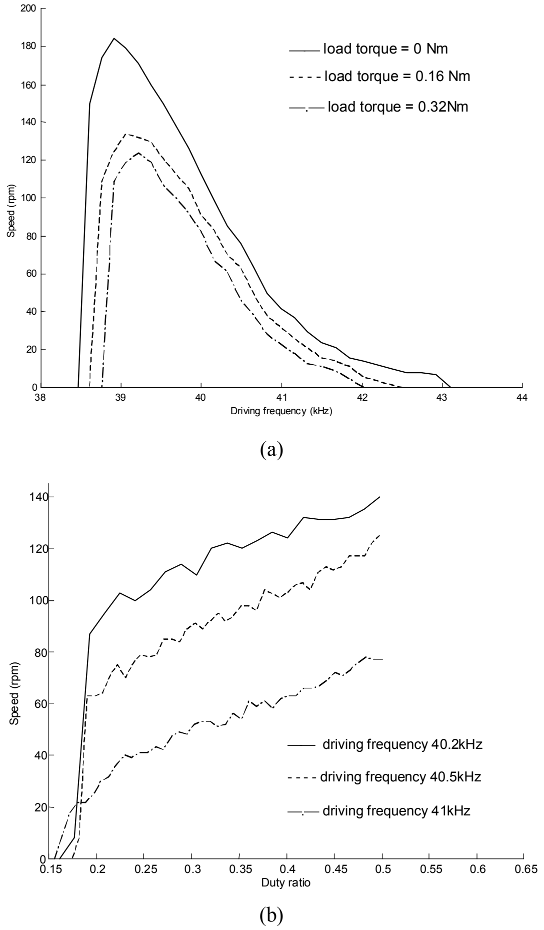

Normal force from spring disc
Normal force from spring disc
Rotor rotates in reverse direction
Rotor
Travelling wave direction
Stator
Piezoelectric elements
Vsinωt
Vsin(ωt+α)
Vsinωt

Fig. 1. Operating principle of TUSM.



(a)

(b)

Fig. 2. Speed characteristics of TUSM. (a) Speed versus driving frequency. (b) Speed versus duty ratio.

This paper introduces neuro-fuzzy speed control of TUSM using two control inputs, namely the frequency and the duty ratio. The speed responses of the TUSM to the frequency and duty ratio facilitate the establishment of the fuzzy rules. As a result, the neuro-fuzzy control at the start gains the human knowledge through its fuzzy logic part. Furthermore, its neural network part can on-line tunes the parameters so that optimal performance can be achieved. The partial linearity of duty ratio also improves the control performance. Finally, different learning rates are adopted at different network nodes to obtain quickest converging time.

In Section II, the operating principle and characteristics of the TUSM will be briefly described. Afterwards, the neuro-fuzzy control structure will be described in Section III. Section IV will be devoted to the system set-up of the proposed servo speed control of the TUSM. The experimental results, and hence the effectiveness of the controller, are then presented and discussed in Section V. Finally, conclusions will be drawn in Section VI.

II. Operating Principle and Motor Characteristics

A TUSM utilizes piezoelectric effect, instead of electromagnetic effect as in traditional electromagnetic motors, to convert electrical energy into mechanical motion. Hence, its operating principle is fundamentally different from that of electromagnetic motors. Fig. 1 illustrates its operating principle. The stator, designed with comb-tooth surface, is attached below the sectors of piezoelectric elements. The piezoelectric elements are divided into sectors instead of a complete ring so that they can be applied with different voltages. The rotor is pressed against the stator by means of a pressure spring disc. It should be noted that the spring as shown is for illustration only. In real case, it is not a normal spring but a pressure disc. Basically, the operating principle of the TUSM is due to mechanical vibrations of piezoelectric elements, as a result of voltages applied on them. Hence they produce a traveling wave in the stator. The surface interaction between the stator and the rotor creates a driving torque in the rotor. As shown in Fig. 1, piezoelectric elements are positioned at an appropriate distance from one another. One fed by the phase A voltage source $V_m \sin\omega t$ to produce a mechanical vibration while another is fed by the phase B voltage source $V_m \sin(\omega t + \alpha)$ to generate another vibration, where V_m is the amplitude, $\omega = 2\pi f$ is the angular version of frequency f and α is the phase difference of the two driving voltage sources. Each voltage source generates a mode of vibration throughout the stator and the superposition of them forms a traveling wave that propagates along the stator. Depending on which phase of the voltage leads the other, the traveling wave will either travel in right-hand direction or in left-hand direction. Elliptical motion can be observed at the wave crests of the surface of the stator. Such motion produces a tangential force at the contact surface between the stator and the rotor. Making use of a spring disc to keep a rigid rotor in pressure contact with the stator, the rotor can be driven to travel in the opposite direction of that of the traveling wave by this tangential force at the contact surface. Although the rotor moves only about a micrometer in an individual cycle of the elliptical motion, this can add up to a speed of several centimeters per second at ultrasonic frequencies. The ring form structure provides floating end condition, or acts as an infinite long beam, for the mechanical vibration, which is essential for the generation of traveling wave. In addition, without excitation by proper electrical power source, the rotor cannot rotate as it is pressed against the stator, hence producing a high holding torque.

Typical nonlinear speed characteristics of the TUSM are illustrated in Fig. 2. From Fig. 2a, it can be seen that the relationship between speed *N* and *f* is highly nonlinear, and the controllable range of *f* is seriously dependent on the load torque. Fig. 2b reveals the speed characteristics of TUSM vs. duty ratio. The speed decreases linearly with decreasing duty ratio. Upon certain threshold value, the motor suddenly halts at certain speed. However, different driving frequencies give different threshold values as well as different rates of change of speed. At each driving frequency, the speed span that the duty ratio control can achieve is pretty limited. The range was roughly 30rpm. As reported in [12], the speed of the TUSM is given by:

$$N = \frac{hk\psi\omega}{2} \tag{1}$$

where k is the angular displacement period of the traveling-wave, h is the thickness of the stator and ψ is the amplitude of the traveling-wave W, which can be expressed as:

$$W = \psi(\omega, k)\sin(\omega t + \varphi) \tag{2}$$

where φ is the phase lag between the sinusoidal voltage sources and the traveling wave. The traveling wave in fact is the vertical deflection of the stator along the neutral axis. From (2), it can readily be observed that ψ is a function of ω and k. It can be solved by the below 2nd order differential equation:

$$\rho A \frac{\partial^2 W}{\partial t^2} + EIk^4 W = -kM_o \sin\omega t \tag{3}$$

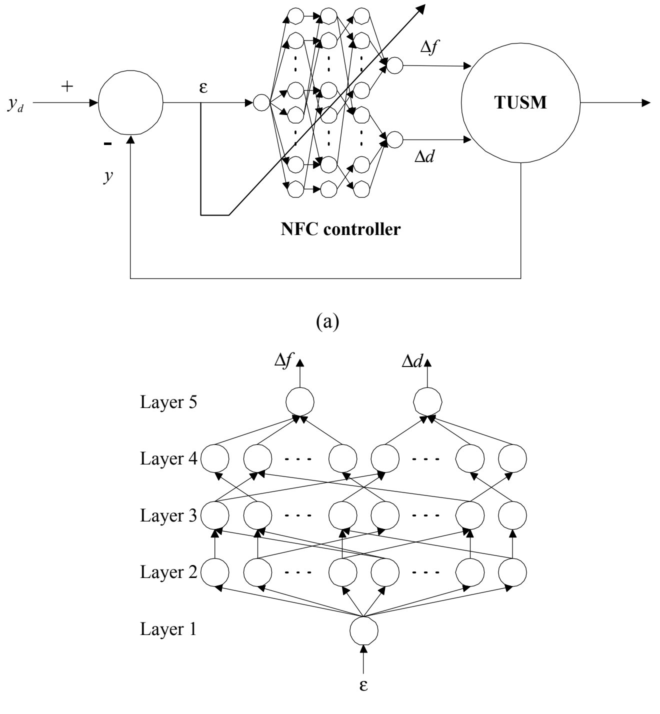
where ρ is the mass per unit volume, A is the cross-sectional area of the stator, E is the Young's modulus of the material and I is the second moment of area. M_o is the moment amplitude resulted from the sinusoidal voltage through piezoelectric effect and is in linear relation with the voltage amplitude. From (3) it can be observed that ψ is directly proportional to M_o. As a result, from (1) and (3), N is in linear relation with the voltage amplitude.

III. Neuro-Fuzzy Controller

Recently, a number of neuro-fuzzy control (NFC) schemes have been proposed, focusing on the variations of structure and parameter learning. In this project, the NFC structure and its mathematical derivation discussed in [20] are adopted. Fig. 3 shows the NFC structure [20] for speed servo control of the TUSM. In Fig. 3a, speed command is inputted to the NFC controller. Error is then obtained by comparing the difference from the actual speed and the speed command. The actual speed is obtained from the optical encoder. The NFC controller then computes the change of frequency and duty ratio required. Since the dynamics of the TUSM is so fast that it is not really necessary to consider the dynamic of this actuator in complete motion control system. As a result, the change of error is not needed. On-line training is also performed by utilizing the error information obtained. As shown in Fig. 3b, in this 5-layered structure, the nodes in the layer 1 (called the input layer) simply direct the crisp input states (namely the speed error ε of the TUSM) to the layer 2. The nodes in the layer 2 are input term nodes representing the input linguistic variables. They act as characteristic functions to transform those crisp values into fuzzy numbers. Hence, fuzzification is performed in this layer. Each node in the layer 3 is a rule node that represents one fuzzy logic rule. As a result, all the layer-3 nodes form a fuzzy rule base. The nodes in the layer 4 are output term nodes representing the output linguistic variables. They serve to integrate the rules which have the same consequence. Finally, the nodes in the layer 5 (called the output layer) convert the resulting fuzzy numbers into the crisp output control signals (Δf and Δd). In other words, the defuzzification is done in this layer.

Based on this NFC structure [20], the net input to the node i in the layer k can be represented by an integration function $F_i^{(k)}(u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}; w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)})$ to combine information from other nodes, where $u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}$ are inputs to this node and $w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}$ are the associated link weights. Then, the output of this node is given by:

$$o_i^{(k)} = a_i^{(k)}(F_i^{(k)}) \tag{4}$$



(a)

(b)

Fig. 3. NFC of TUSM. (a) System configuration. (b) Controller structure.

where $a_i^{(k)}$ is the activation function of the node *i* in the layer *k*. The following derivations are mainly based on the NFC approach discussed in [20].

As each node in the layer 1 simply directs the crisp input value to the layer 2, the output $o_i^{(1)}$ is identical to its input $u_i^{(1)}$, yielding:

$$F_i^{(1)} = u_i^{(1)} \tag{5}$$

$$a_i^{(1)} = F_i^{(1)} \tag{6}$$

Hence, the layer-1 link weight $w_i^{(1)}$ is always unity. In the proposed application, $u_1^{(1)}$ is ε.

In the layer 2, the fuzzification process transforms a crisp input $u_i^{(2)}$ into a fuzzy output $o_i^{(2)}$ which is characterized by its degree of belonging to the *j*th term of the *i*th input linguistic variable x_i, based on the user-specified membership function $M_{x_i}^j$. When a bell-shaped characteristic function is adopted [22], the fuzzified output in this layer is given by:

$$o_i^{(2)} = M_{x_i}^j\left(u_i^{(2)}, m_{ij}^{(2)}, \sigma_{ij}^{(2)}\right) = \exp\left(-\frac{\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^2}\right) \tag{7}$$

where $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$ are the mean and the variance of the characteristic function, respectively. Hence, the corresponding integration and activation functions can be expressed as:

$$F_i^{(2)} = -\frac{\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^2} \tag{8}$$

$$a_i^{(2)} = \exp\left(F_i^{(2)}\right) \tag{9}$$

Thus, the layer-2 link weight $w_{ij}^{(2)}$ is governed by both $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$. In this application, the input linguistic variables of ε are described by bell-shaped characteristic functions.

As the links in the layer 3 are used to perform precondition matching of fuzzy logic rules, those rule nodes carry out the fuzzy AND operation. However, for speed control of TUSM, the rules, which are tabulated in Table 1, are very simple. They are established according to the characteristics of the TUSM and the should-be behavior of a motor controller. As a result, the fuzzy input $u_p^{(3)}$ can be directly mapped to the fuzzy output $o_i^{(3)}$ with reference to the table. Hence, it yields:

$$F_i^{(3)} = u_p^{(3)} \tag{10}$$

$$a_i^{(3)} = F_i^{(3)} \tag{11}$$

The layer-3 link weight $w_i^{(3)}$ is then always unity. It should be noted that for each rule node, there is at most one link (maybe none) from those term nodes of a linguistic node. By describing rules in a human way, the nonlinear relationship between (Δ*f*, Δ*d*) and ε can be easily embedded into the controller structure.

In the layer 4, the links perform the fuzzy OR operation to integrate the rules which have the same consequence. The fuzzy output $o_i^{(4)}$ can thus be evaluated as the summation of those fuzzy inputs $u_1^{(4)}, u_2^{(4)}, \ldots, u_p^{(4)}$, but bounded by unity. Hence, the corresponding integration and activation functions are given by:

$$F_i^{(4)} = \sum_i u_i^{(4)} \tag{12}$$

$$a_i^{(4)} = \min(1,\ F_i^{(4)}) \tag{13}$$

Also, the layer-4 link weight $w_i^{(4)}$ is always unity.

Finally, the linguistic control signals Δ*f* and Δ*d* in the layer 5 need to be converted back to the crisp values. By adopting the center-of-area defuzzification method [20], the defuzzified output is given by:

$$o_i^{(5)} = \frac{\sum_j \left(m_{ij}^{(5)}\sigma_{ij}^{(5)}\right)u_{ij}^{(5)}}{\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}} \tag{14}$$

where $u_{ij}^{(5)}$ is the input of the *j*th term of the *i*th output linguistic variable, while $m_{ij}^{(5)}$ and $\sigma_{ij}^{(5)}$ are the corresponding mean and variance of the bell-shaped membership function, respectively [22]. Hence, it yields:

$$F_i^{(5)} = \sum_j \left(m_{ij}^{(5)}\sigma_{ij}^{(5)}\right)u_{ij}^{(5)} \tag{15}$$

$$a_i^{(5)} = \frac{F_i^{(5)}}{\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}} \tag{16}$$

Thus, the layer-5 link weight $w_{ij}^{(5)}$ is $m_{ij}^{(5)}\sigma_{ij}^{(5)}$. In this application, the bell-shaped characteristic functions are used to describe the output linguistic variables Δ*f* and Δα.

Therefore, this NFC structure possesses the features of fuzzy logic control due to the operation of linguistic variables, characteristic functions and fuzzy control rules, as well as the symptoms of neural network control due to the structure involving nodes, layers and weights.

Having established the NFC structure, the network enters the learning phase to adjust the parameters of the input and output membership functions optimally. The back-propagation algorithm is adopted for this supervised learning. The goal is to minimize the error function *E* as given by:

$$E = \frac{1}{2}\left(y^d(t) - y(t)\right)^2 = \frac{1}{2}\varepsilon^2 \tag{17}$$

TABLE I. FUZZY CONTROL RULES

ε	NB	NM	NS	ZO	PS	PM	PB
Δf	PB	PM	PS	ZO	NS	NM	NB
Δd	NB	NM	NS	ZO	PS	PM	PB

where $y^d(t)$ and $y(t)$ represent the desired and actual position outputs of the TUSM, respectively. Then, starting at the output nodes, a backward pass is used to compute $\partial E/\partial w$ for all the hidden nodes. The corresponding general learning rule [22], [23] is expressed as:

$$w(t+T) = w(t) + \eta\left(-\frac{\partial E}{\partial w}\right) \tag{18}$$

where η is the learning rate and T is the interval for successive update.

On the other hand, this NFC structure suffers from a problem that it is difficult to relate the motor output error to the controller output error, due to the absence of the mathematical model and hence the Jacobian of the USM. Although an additional neural network, or neuro-fuzzy network, can be incorporated to estimate the Jacobian, it requires heavy computational effort and suffers from difficulty in implementation. In order to solve the problem because of the unknown Jacobian, an approximation of $\partial E/\partial w$ can be employed [25] as given by:

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial p}\frac{\partial p}{\partial w} \approx -\varepsilon \ \mathrm{sgn}\left(\frac{\partial y}{\partial p}\right)\frac{\partial p}{\partial w} \tag{19}$$

where p is the controller output or USM input (namely Δf or Δd). Notice that $\partial p/\partial w$ can be readily obtained through back-propagation, while $\mathrm{sgn}(\partial y/\partial p)$ is generally known during operation. The following derivations are mainly based on the NFC learning approach discussed in [22].

In the layer 5, $w_{ij}^{(5)}$ is $m_{ij}^{(5)}\sigma_{ij}^{(5)}$. The adaptive rules of $m_{ij}^{(5)}$ and $\sigma_{ij}^{(5)}$ are derived as:

$$\frac{\partial E}{\partial m_{ij}^{(5)}} = \frac{\partial E}{\partial a_i^{(5)}}\frac{\partial a_i^{(5)}}{\partial m_{ij}^{(5)}} \tag{20}$$

$$\frac{\partial E}{\partial \sigma_{ij}^{(5)}} = \frac{\partial E}{\partial a_i^{(5)}}\frac{\partial a_i^{(5)}}{\partial \sigma_{ij}^{(5)}} \tag{21}$$

Substituting $y(t) = a_i^{(5)}$ into (17), $\partial E/\partial a_i^{(5)}$ can be obtained as:

$$\frac{\partial E}{\partial a_i^{(5)}} = -\left(y^d(t) - y(t)\right) = -\varepsilon \tag{22}$$

By using (12) and (13), $\partial a_i^{(5)}/\partial m_{ij}^{(5)}$ can be obtained as:

$$\frac{\partial a_i^{(5)}}{\partial m_{ij}^{(5)}} = \frac{\sigma_{ij}^{(5)}u_{ij}^{(5)}}{\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}} \tag{23}$$

Hence, by substituting (22) and (23) into (20), and then using (18), $m_{ij}^{(5)}$ is updated by:

$$m_{ij}^{(5)}(t+T) = m_{ij}^{(5)}(t) + \eta\varepsilon\frac{\sigma_{ij}^{(5)}u_{ij}^{(5)}}{\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}} \tag{24}$$

Similarly, by using (15), (16), (18), (21) and (22), $\sigma_{ij}^{(5)}$ is updated by:

$$\sigma_{ij}^{(5)}(t+T) = \sigma_{ij}^{(5)}(t) + \eta\varepsilon\frac{m_{ij}^{(5)}u_{ij}^{(5)}\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)} - u_{ij}^{(5)}\sum_j m_{ij}^{(5)}\sigma_{ij}^{(5)}u_{ij}^{(5)}}{\left(\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}\right)^2} \tag{25}$$

By using (22), the error $\delta_i^{(5)}$ to be propagated to the layer 4 is given by:

$$\delta_i^{(5)} = -\frac{\partial E}{\partial a_i^{(5)}} = \varepsilon \tag{26}$$

Since $w_i^{(4)}$ in the layer 4 is always unity, no weight adjustment is needed. Nevertheless, the error $\delta_i^{(4)}$ needs to be computed and propagated as given by:

$$\delta_i^{(4)} = -\frac{\partial E}{\partial a_i^{(4)}} \tag{27}$$

Since $a_i^{(4)} = u_{ij}^{(5)}$, (27) can be rewritten as:

$$\delta_i^{(4)} = -\frac{\partial E}{\partial a_i^{(5)}}\frac{\partial a_i^{(5)}}{\partial u_{ij}^{(5)}} \tag{28}$$

By using (15) and (16), $\partial a_i^{(5)}/\partial u_{ij}^{(5)}$ can be obtained. Hence, making use of (26), (28) can be expressed as:

$$\delta_i^{(4)} = \varepsilon\frac{m_{ij}^{(5)}\sigma_{ij}^{(5)}\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)} - \sigma_{ij}^{(5)}\sum_j m_{ij}^{(5)}\sigma_{ij}^{(5)}u_{ij}^{(5)}}{\left(\sum_j \sigma_{ij}^{(5)}u_{ij}^{(5)}\right)^2} \tag{29}$$

Similar to the layer 4, there is no weight to be adjusted in the layer 3. By employing $a_i^{(3)} = u_i^{(4)}$, $a_i^{(4)} = u_i^{(4)}$ and (27), the error $\delta_i^{(3)}$ can be easily obtained as:

$$\delta_i^{(3)} = -\frac{\partial E}{\partial a_i^{(3)}} = -\frac{\partial E}{\partial a_i^{(4)}}\frac{\partial a_i^{(4)}}{\partial u_i^{(4)}} = \delta_i^{(4)} \tag{30}$$

In the layer 2, similar to the layer 5, the adaptive rules of $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$ are derived as:

$$\frac{\partial E}{\partial m_{ij}^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}}\frac{\partial a_i^{(2)}}{\partial m_{ij}^{(2)}} \tag{31}$$

$$\frac{\partial E}{\partial \sigma_{ij}^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}}\frac{\partial a_i^{(2)}}{\partial \sigma_{ij}^{(2)}} \tag{32}$$

Since $a_i^{(2)} = u_i^{(3)}$ and using (5.36), the error $\delta_i^{(2)}$ is given by:

$$\delta_i^{(2)} = -\frac{\partial E}{\partial a_i^{(2)}} = -\frac{\partial E}{\partial u_i^{(3)}} = -\frac{\partial E}{\partial a_i^{(3)}}\frac{\partial a_i^{(3)}}{\partial u_i^{(3)}} = \delta_i^{(3)}\frac{\partial a_i^{(3)}}{\partial u_i^{(3)}} \tag{33}$$

From (10) and (11), $\partial a_i^{(3)} / \partial u_i^{(3)}$ can be obtained. Hence, (33) can be rewritten as:

$$\delta_i^{(2)} = \delta_i^{(3)} \tag{34}$$

By using (8) and (9), $\partial a_i^{(2)} / \partial m_{ij}^{(2)}$ can be obtained as:

$$\frac{\partial a_i^{(2)}}{\partial m_{ij}^{(2)}} = \exp\left(-\frac{\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^2} \right) \frac{2\left(u_i^{(2)} - m_{ij}^{(2)}\right)}{\left(\sigma_{ij}^{(2)}\right)^2} \tag{35}$$

Hence, by substituting (32) and (33) into (31), and then using (18), $m_{ij}^{(2)}$ is updated by:

$$m_{ij}^{(2)}(t+T) = m_{ij}^{(2)}(t) + \eta \delta_i^{(2)} \exp\left(-\frac{\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^2} \right) \frac{2\left(u_i^{(2)} - m_{ij}^{(2)}\right)}{\left(\sigma_{ij}^{(2)}\right)^2} \tag{36}$$

Similarly, by using (8), (9), (18), (32) and (33), the update rule of $\sigma_{ij}^{(2)}$ becomes:

$$\sigma_{ij}^{(2)}(t+T) = \sigma_{ij}^{(2)}(t) + \eta \delta_i^{(2)} \exp\left(-\frac{\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^2} \right) \frac{2\left(u_i^{(2)} - m_{ij}^{(2)}\right)^2}{\left(\sigma_{ij}^{(2)}\right)^3} \tag{37}$$

Therefore, based on (24), (25), (36) and (37), those weights described by $m_{ij}^{(2)}$, $\sigma_{ij}^{(2)}$, $m_{ij}^{(5)}$ and $\sigma_{ij}^{(5)}$ are continually updated by back-propagation. To further improve the learning speed, different learning rates are used at different nodes. This is different from the traditional approach [4], [22], which employs the same learning rate throughout the network. In this paper, 0.005 is used for all the nodes dealing with defuzzification of duty ratio whereas 0.001 is used for all the nodes dealing with defuzzification of frequency and fuzzification of error. The rates are determined by trial and error such that optimal results are obtained for square-wave command. It should be noted that the corresponding convergence speed is superior to that of the normal back-propagation scheme for neural networks because expertise has been implanted as characteristic functions and fuzzy rules before the learning phase.

IV. Speed Control System

Our previous paper [10] adopts a general servo control system for hardware implementation. The system generally employs a microcomputer which commands a dual-channel function generator to provide proper switching signals for the 2-phase inverter. However, this system suffers from three key shortcomings which make it unrealistic for practical application. First, the general purpose interface bus (GPIB), which links between the microcomputer and the function generator, is not fast enough to provide good dynamic performance. It is acceptable in case of position control, as in our previous paper [10]. Yet such system configuration is not applicable in servo speed control. Second, it is impractical to dedicate a PC to a TUSM. Third, the required dual-channel function generator, having internal synchronization capability, is expensive and bulky.

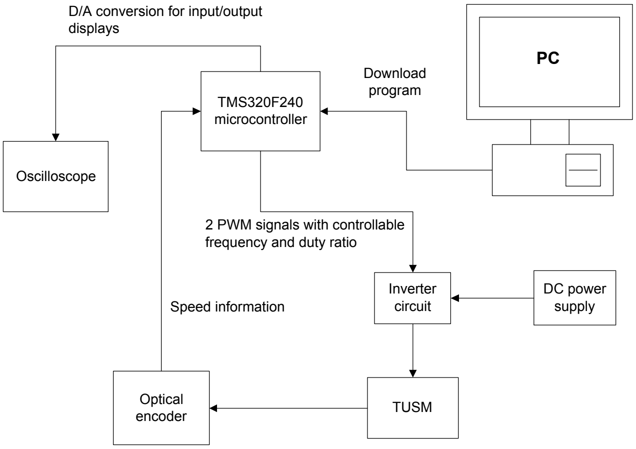

D/A conversion for input/output displays
PC
Download program
TMS320F240 microcontroller
Oscilloscope
2 PWM signals with controllable frequency and duty ratio
Speed information
Inverter circuit
DC power supply
Optical encoder
TUSM

Fig. 4. The block diagram of the TUSM speed control system.

Since the operating frequency of the TUSM is over 40 kHz, conventional microcontrollers are not capable for the task. Thus, a new DSP microcontroller TMS320F240 of Texas' Instruments (TI) is selected as the core of hardware implementation. By using this TMS320F240, the sampling frequency for speed control can be greatly increased; hence the dynamic performance can be significantly improved.

Fig. 4 shows the system control hardware for speed control of the TUSM. The control software is written in computer programming language C and assembly language. Afterwards, the programs are compiled and linked in an IBM PC. Then, the executable is downloaded into the TMS320F240. After downloading, it can run in the standalone mode. This TMS320F240 is externally driven by a 20MHz crystal, which is capable to generate high-frequency PWM signals with variable phase differences, variable frequencies and variable duty ratios. Since digital PWM signals are directly outputted to the 2-phase inverter circuit, the electrical noise immunity and the control reliability are much better. Also, its built-in edge-signal detection feature allows it to directly gather data from the optical encoder without signal conditioning. Furthermore, by adding a simple phase split circuit, one TMS320F240 can control up to two TUSMs simultaneously, leading to favor its application to mechatronics and robotics.

Fig. 5 shows the 2-phase inverter circuit for the TUSM. It adopts an H-bridge configuration to provide the PWM switching voltages. Since the input terminals of the TUSM are inherently capacitive, two small inductors L1 and L2, of value 1mH, are purposely added in series to its terminals so that resonance is resulted. The corresponding resonance can offer an advantageous feature of soft switching, hence reducing the switching loss and electromagnetic interference. Those power devices connecting to the positive rail are p-channel IRF9640 MOSFET and those to the negative rail are n-channel IRF740 MOSFET. Such arrangement can minimize the required number of isolated DC supplies for MOSFET gate driving.

For the sake of comparison, a dual-mode PI speed controller is also employed. The PI speed controller is similar to [12], except the sampling time is the same as the NFC controller. The speed error ε is obtained by comparing the speed command and the speed feedback outputted by the optical encoder. Afterwards, it is fed into the PI controller which generates a control signal *g*. Through a mapping function, the actual control variables, namely the duty ratio *d* and the frequency *f*, are found. The inverter circuit changes accordingly and powers the TUSM.

The mapping function which maps *g* to *f* and *d* is expressed as:

$$\begin{cases} f = \mathrm{F}(g) \\ d = \mathrm{D}(g, f) \end{cases} \tag{38}$$

The functions F and D are given by:

$$\mathrm{F}(g) = \begin{cases} f_0 & \text{if } 0 < g \le g_0 \\ f_1 & \text{if } g_0 < g \le g_1 \\ \vdots & \quad \vdots \\ f_n & \text{if } g \ge g_{n-1} \end{cases} \tag{39}$$

$$\mathrm{D}(g, f) = \mathrm{A}(f)g + \mathrm{B}(f) \tag{40}$$

where A and B are the slope and the intercept of the line equation, respectively. Both A and B depend on the driving frequency. For speed control, 5 values of driving frequency were chosen so that minimum number of driving frequency is guaranteed to span the whole speed range of the TUSM. The 5 values are 40.16, 40.32, 40.65, 40.82, 41.15 kHz.

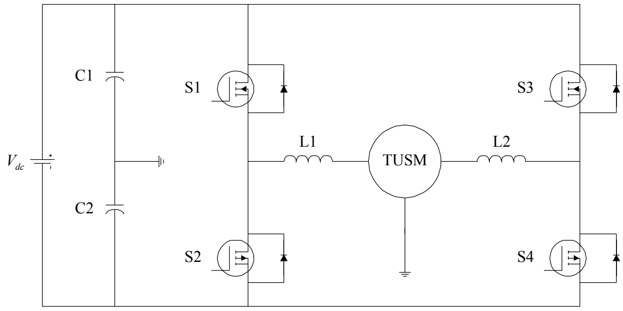


Fig. 5. Schematic of 2-phase inverter circuit.

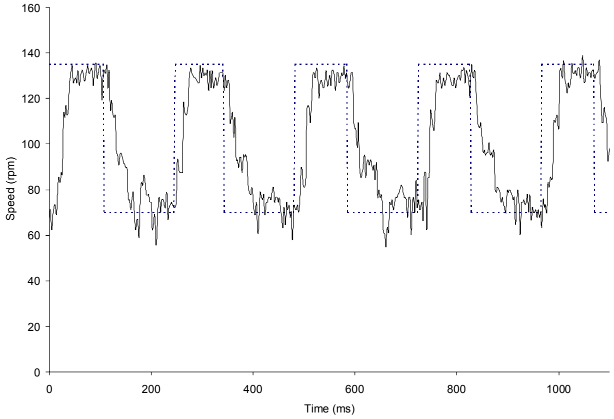


Fig. 6. Speed response by dual-mode PI control under square-wave command.

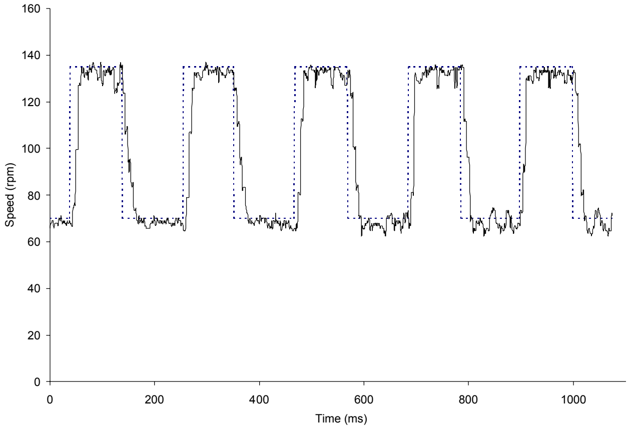


Fig. 7. Speed response by NFC controller to square-wave command.

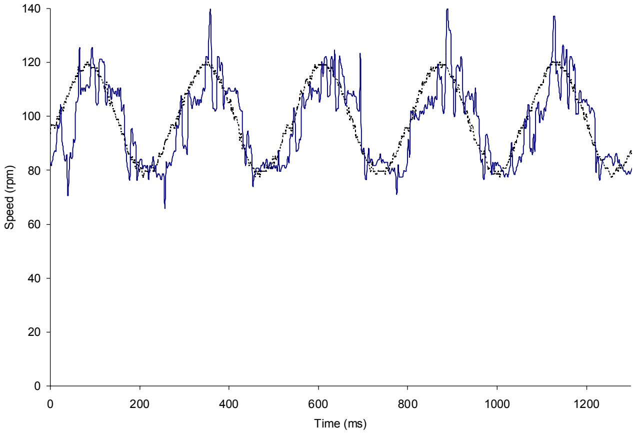


Fig. 8. Speed response by PI controller under sine-wave command.

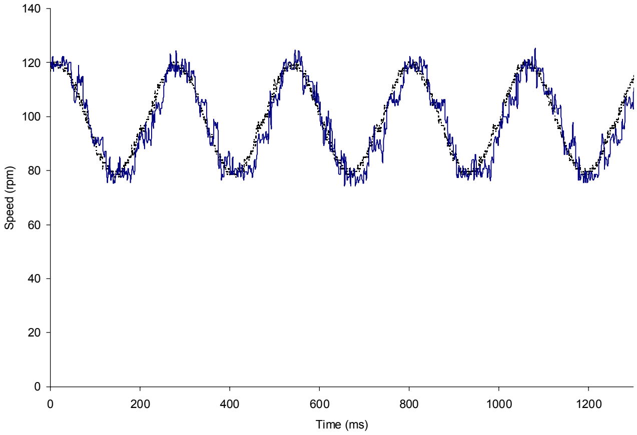


Fig. 9. Speed responses by NFC controller under sine-wave command.

Since there is no applicable mathematical model of the TUSM, the PI controller parameters are chosen by trial and error such that optimal performance occurs at testing condition. In this case, the PI constants are selected as K_p = 0.05 and K_i = 4. The sampling time for both controllers is set as 10ms. Motor specifications are listed in Table 2.

TABLE II. TUSM SPECIFICATION

Rated speed (rpm)	90-100
Rated torque (Ncm)	32
Supply voltage range (V)	100-130
Supply frequency range (kHz)	40-44

V. EXPERIMENTAL RESULTS

Fig. 6 shows the speed responses of the TUSM under dual-mode PI control. It was under no load condition. The dotted line indicates the input command reference while the solid line refers to the actual motor speed. It can be observed that though the PI speed controller manages to reduce the steady-state error at high speed, it performs poorly at low speed. The controller also took 40ms to reach the command. speed. Furthermore, the speed ripple can also be found at both high level and low level. The ripple was particularly significant at low speed level. In contrast, the NFC controller had much better response upon learning. Fig. 7 shows the speed response after 1000 cycles. It only took 20ms to reach the command speed. Furthermore, there was very little oscillation at both high and low speed levels, contrasting to that of the PI controller.

To demonstrate the flexibility of the NFC controller, the control system was also subject to a sinusoidal-wave command. As shown in Fig. 8, the PI controller performed poorly. Significant spikes can be observed at the peaks. Oscillations appeared throughout. The phase lag was also noticeable.

On the other hand, the NFC controller gave much better results, as shown in Fig. 9. Upon learning, the NFC controller successfully tracked the command closely, though the learning time increased from 1000 cycles to 4500 cycles. It indicates that the initial parameters of the fuzzy logic control match closer to square-wave command than that to sinusoidal-wave command.

VI. CONCLUSIONS

In this paper, a NFC controller has been successfully developed for servo speed control of the TUSM. It takes the advantages of the human-knowledge-incorporated fuzzy logic and the self-adaptive neural network. Furthermore, using frequency and duty ratio as control variables makes the control system more practical and efficient. Experimental results demonstrate the superiority of the NFC controller over the dual-mode PI controller. Its adaptability to various command patterns can readily be observed.

ACKNOWLEDGMENT

This work was supported and funded by the Hong Kong Research Grants Council (Project No. HKU 7039/00E).

REFERENCES

[1] S. I. Furuya, Y. Ohkura, and T. Maruhashi, "A novel inverter-drive ultrasonic motor-actuated positioning servo motion system using a fuzzy reasoning control scheme," *Journal of Circuits, Systems and Computers*, Vol. 4, No. 4, 1994, pp. 395-414.

[2] T. Senjyu, and K. Uzeato, "Adjustable speed control of ultrasonic motors by adaptive control," *IEEE Power Electronics Specialists Conference*, Vol. 2, 1994, pp. 1237-1242.

[3] T. Senjyu, H. Miyazato, and K. Uzeato, "Position control of ultrasonic motors using neural network," *Proceeding of IEEE International Industrial Electronics Symposium*, Vol. 1, 1996, pp. 368-373.

[4] T. Senjyu, H. Miyazato, and K. Uzeato, "Speed control of ultrasonic motors using fuzzy neural network," *Proceedings of IEEE International Power Electronics Congress*, 1996, pp. 29-34.

[5] T. Senjyu, H. Miyazato, S. Yokoda, and K. Uzeato, "Speed control of ultrasonic motors using neural network," *IEEE Transactions on Power Electronics*, Vol. 13, No. 3, 1998. pp. 381-387.

[6] Y. Izuno, and M. Nakaoka, "High performance and high precision ultrasonic motor-actuated positioning servo drive system using improved fuzzy-reasoning controller," *IEEE Power Electronics Specialists Conference*, Vol. 2, 1994, pp. 1269-1274.

[7] Y. Izuno, M. Hojo, and M. Nakaoka, "Software-based adjusting P-I gain controller for speed-tracking servo system using traveling-wave type ultrasonic motor," *Industry Applications Society Annual Meeting*, Vol. 3, 1994, pp. 1777-1784.

[8] F.J. Lin, "Fuzzy adaptive model-following position control for ultrasonic motor," *IEEE Transactions on Power Electronics*, Vol. 12, No. 2, 1997, pp. 261-268.

[9] F.J. Lin, W.J. Hwang, and R.J. Wai, "Ultrasonic motor servo-drive with online trained neural-network model-following controller," *IEE Proceedings – Electric Power Applications*, Vol. 145, 1998, pp. 105-110.

[10] K.T. Chau, and S.W. Chung, "Servo position control of ultrasonic motors using fuzzy neural network," *Electric Machines and Power Systems*, Vol. 29, No. 3, 2001, pp. 229-246.

[11] T. Senjyu, H. Miyazato, and K. Uzeato, "Quick and precise position control of an ultrasonic motor with dual mode control," *International Journal of Electronics*, Vol. 80, No. 2, 1996, pp. 191-200.

[12] K.T. Chau, and S.W. Chung, "Servo Speed Control of Traveling-Wave Ultrasonic Motors using Pulse Width Modulation," *Electric Power Components and Systems*, Vol. 29, No. 8, 2001, pp. 707-722.

[13] K. Nishibori, S. Kondo, H. Obata, and S. Okuma, "PWM driving characteristics of robot hand with fingers using vibration-type ultrasonic motors," *IEEE Industrial Electronics Conference*, Vol. 3, 1997, pp. 1355-1360.

[14] P. Hagedorn, and J. Wallaschek, "Traveling wave ultrasonic motors, part I: working principle and mathematical modeling of the stator," *Journal of Sound and Vibration*, Vol. 155, No. 1, 1992, pp. 31-46.

[15] N.W. Hagood IV, and A.J. McFarland, "Modeling of a piezoelectric rotary ultrasonic motor," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, Vol. 42, No. 2, 1995, pp. 210-224.

[16] J. Mass, and H. Grotstollen, "Averaged model of inverter-fed ultrasonic motors," *Proceeding of IEEE Power Electronics Specialists Conference*, Vol. 1, 1997, pp. 740-746.

[17] F.J. Lin, "Fuzzy adaptive model-following position control for ultrasonic motor," *IEEE Transactions on Power Electronics*, Vol. 12, No. 2, 1997, pp. 261-268.

[18] F.J. Lin, R.F. Fung, and R.J. Wai, "Comparison of sliding-mode and fuzzy neural network control for motor-toggle servomechanism," *IEEE/ASME Transactions on Mechatronics*, Vol. 3, No. 4, 1998, pp. 302-318.

[19] F.J. Lin, R.J. Wai and R.Y. Duan, 1999, "Fuzzy Neural networks for identification and control of ultrasonic motor drive with LLCC resonant technique," *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 5, 1999, pp. 999-1011.

[20] C. T. Lin, and C.S.G., Lee, *Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, 1996.

[21] P. M. Mills, A. Y. Zomaya and M. O. Tadè, *Neuro-Adaptive Process Control A Practical Approach*. John Wiley & Sons, 1996.

[22] Y. Zhang, P. Sen, and G.E. Hearn, "An on-line trained adaptive neural controller," *IEEE Control Systems Magazine*, Vol. 15, No. 5, 1995, pp. 67-75.