

# Neural Network Compensation of Gear Backlash Hysteresis in Position-Controlled Mechanisms

David R. Seidl, Sui-Lun Lam, Jerry A. Putman, and Robert D. Lorenz, *Senior Member, IEEE*

**Abstract**—This paper demonstrates that artificial neural networks can be used to identify and compensate for hysteresis caused by gear backlash in precision position-controlled mechanisms. A major contribution of this research is that physical analysis of the system nonlinearities and optimal control are used to design the neural network structure. Network sizing and initializing problems are thus eliminated. This physically meaningful, modular approach facilitates the integration of this neural network with existing controllers; thus, initial performance matches that of existing control approaches and then is improved by refining the parameter estimates via further learning. The neural network operates by recognizing backlash and switching to a control which moves smoothly through the backlash when the torque transmitted to the output shaft must be reversed.

## I. INTRODUCTION

THE hysteresis caused by gear backlash is a well-understood dynamic nonlinearity. Discussions can be found in several control texts [1]–[2]. The standard linear feedback (PD or PID) controller ignores the dynamic properties of backlash leading to tracking errors. Specifically,

- the gears must be engaged in the correct direction before transmitting torque;
- the motor acts on only the motor shaft—not the lumped—dynamics when reversing direction; and
- the engaging of the gears is an inelastic collision which, because of its impulsive nature, can excite otherwise benign high-frequency resonances if not limited.

Moreover, overshoot caused by improperly referenced, underdamped, or integral error feedback exacerbates these problems, forcing torque reversals. These deficiencies lead to high-frequency limit cycles (high controller gains) or poor disturbance rejection (low controller gains).

An adaptive exact inverse backlash model controller was proposed in [3]. The exact inverse requires creating an ideal impulse in the motor shaft velocity to achieve an instantaneous

move from one end of the backlash region to the other. This is not realizable since the motor shaft position cannot change instantaneously. The exact inverse does however provide the mathematically ideal control for which we seek the best possible approximation.

The solution examined in this paper uses the desired load shaft trajectory to create corresponding motor shaft and motor torque trajectories. First, the gear torque needed to follow the desired load shaft trajectory is computed; it includes inertia and friction decoupling and position and velocity error feedback. Then a reference trajectory for the motor shaft is created that moves the motor shaft to the correct backlash boundary in a time-optimal fashion (subject to relative acceleration limits) and maintains it there until the desired gear torque is reversed. A torque command for the motor shaft is then created that decouples the motor shaft inertia and friction, provides velocity and position error feedback and, if the gear is engaged, supplies the desired gear torque. Proper feedforward and references are used, and feedback is tuned to avoid overshoot and gear reversals. Avoiding reversals allows the load inertia and friction to hold the shaft against the gear resulting in improved tracking.

This controller's need for fast nonlinear computing and parameter identification suggests the use of a neural network. Neural networks have several useful attributes:

- They can approximate a nonlinear function as accurately as desired if enough neurons are available and the correct weights can be found.
- They can compute quickly (propagation delays in the tens of nanoseconds) when implemented in VLSI due to their structure of simple components in parallel.
- They can be adapted (learn) with gradient-descent, least squared-error algorithms.

However, standard neural network development procedures also have significant drawbacks:

- The number of neurons needed and their proper distribution into layers are unknown, requiring experimentation with different configurations until satisfactory results are obtained.
- The gradient-descent weight adaptation algorithms used to iteratively improve the network's approximation of the ideal controller can be slow and stall in poor solutions, particularly when more than one hidden layer is used.
- The adaptation procedure is initialized by random weights which disregard existing system knowledge.
- The resulting network yields no insight into how it works, providing no way to adapt it to related problems.

Paper MSDAD 95-18, approved by the Industrial Automation and Control Committee of the IEEE Industry Applications Society for presentation at the 1993 IEEE Industry Applications Society Annual Meeting, Toronto, Ont., Canada, October 3-8. This work received support from the Wisconsin Electric Machines and Power Electronics Consortium (WEMPEC) of the University of Wisconsin-Madison, the National Science Foundation (NSF) funded Laboratory for Applied Manufacturing Controls (LAMC), and the Rockwell Foundation. Manuscript released for publication June 15, 1995.

D. R. Seidl is with UNICO, Franksville, WI 53126 USA.

S.-L. Lam was with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA. He is now with Secure Medical Products Inc., Whitewater, WI 53190 USA.

J. A. Putnam and R. D. Lorenz are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA.  
IEEE Log Number 9414445.

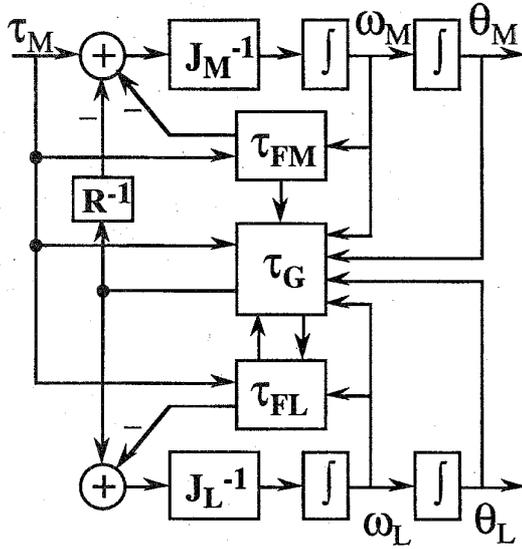


Fig. 1. The backlash model.

These problems can be minimized or avoided by directly constructing the neural network to implement a desirable control algorithm. The algorithm is obtained using state space, nonlinear decoupling, and optimal control techniques. The basic operation of the neuron makes it ideal for integrating state error feedback with the switching surfaces produced by time-optimal or sliding mode control algorithms. The neural network backlash controller developed in this paper demonstrates the use of neurons to implement:

- linear state error feedback,
- time optimal switching curves, and
- conditional terms to decouple nonlinear friction.

This paper presents the backlash model and derives the control algorithm. Then, neural network basics and construction techniques are discussed and applied to the controller. Finally, experimental results are presented.

## II. DETAILS OF THE BACKLASH MODEL

The state equations describing backlash hysteresis are given by (1). The subscripts  $M$  and  $L$  denote motor and load shaft quantities. The time dependent variables are angular acceleration ( $\alpha$ ), velocity ( $\omega$ ), position ( $\theta$ ), and torque ( $\tau$ ). The torque terms are the motor air gap torque ( $\tau_M$ ), the friction torques ( $\tau_{FM}$  and  $\tau_{FL}$ ) and the torque transmitted through the gear to the load shaft ( $\tau_G$ ). The physical constants are the inertias ( $J_M$  and  $J_L$ ), the viscous ( $B_M$  and  $B_L$ ), coulomb ( $C_M$  and  $C_L$ ), and static ( $S_M$  and  $S_L$ ) friction values, the gear ratio ( $R$ ) and half of the angular deadzone distance between gears on the motor shaft side ( $\sigma$ ). The friction torques (1e, f) and gear transmission torque (1g) are functions of the state ( $\omega_M, \theta_M, \omega_L$ , and  $\theta_L$ ) and the input ( $\tau_M$ ) and receive their time dependence through these variables. The time dependence and state dependence designations are omitted for conciseness except when needed. Time instants just before and after time

$t$  are denoted by  $t^-$  and  $t^+$

$$\frac{d\omega_M}{dt} = J_M^{-1}[-\tau_{FM} - R\tau_G + \tau_M] \quad (1a)$$

$$\frac{d\theta_M}{dt} = \omega_M \quad (1b)$$

$$\frac{d\omega_L}{dt} = J_L^{-1}[-\tau_{FL} + \tau_G] \quad (1c)$$

$$\frac{d\theta_L}{dt} = \omega_L \quad (1d)$$

where

$$\begin{aligned} \tau_{FM} = & -B_M\omega_M - C_M\text{sgn}(\omega_M) \\ & - \text{sgn}(\tau_M - R\tau_G) \min(|S_M|, |\tau_M - R\tau_G|) 1\{\omega_M = 0\} \end{aligned} \quad (1e)$$

$$\begin{aligned} \tau_{FL} = & -B_L\omega_L - C_L\text{sgn}(\omega_L) \\ & - \text{sgn}(-\tau_G) \min(|S_L|, |-\tau_G|) 1\{\omega_L = 0\} \end{aligned} \quad (1f)$$

$$\tau_G = \tau_D + \tau_{PM} + \tau_{NM} + \tau_{PS} + \tau_{NS} + \tau_{PI} + \tau_{NI} \quad (1g)$$

$$\tau_D = 0 \cdot 1\{-\sigma < \theta_M - R^{-1}\theta_L < \sigma\} \quad (1h)$$

$$\begin{aligned} \tau_{PM} = & \max(0, (J_M + R^2J_L)^{-1}[RJ_L(\tau_M - \tau_{FM}) + J_M\tau_{FL}]) \\ & \cdot 1\{\theta_M - R^{-1}\theta_L = \sigma \cap \omega_M = R^{-1}\omega_L \neq 0\} \end{aligned} \quad (1i)$$

$$\begin{aligned} \tau_{NM} = & \min(0, (J_M + R^2J_L)^{-1}[RJ_L(\tau_M - \tau_{FM}) + J_M\tau_{FL}]) \\ & \cdot 1\{\theta_M - R^{-1}\theta_L = -\sigma \cap \omega_M = R^{-1}\omega_L \neq 0\} \end{aligned} \quad (1j)$$

$$\begin{aligned} \tau_{PS} = & \max(0, \tau_M - S_M) \\ & \cdot 1\{\theta_M - R^{-1}\theta_L = \sigma \cap \omega_M = R^{-1}\omega_L = 0\} \end{aligned} \quad (1k)$$

$$\begin{aligned} \tau_{NS} = & \min(0, \tau_M + S_M) \\ & \cdot 1\{\theta_M - R^{-1}\theta_L = -\sigma \cap \omega_M = R^{-1}\omega_L = 0\} \end{aligned} \quad (1l)$$

$$\begin{aligned} \tau_{PI} = & J_M J_L (J_M + R^2 J_L)^{-1} (R\omega_M(t^-) - \omega_L(t^-)) \\ & \cdot \delta\{\theta_M - R^{-1}\theta_L = \sigma \cap \omega_M(t) > R^{-1}\omega_L(t)\} \end{aligned} \quad (1m)$$

$$\begin{aligned} \tau_{NI} = & J_M J_L (J_M + R^2 J_L)^{-1} (R\omega_M(t^-) - \omega_L(t^-)) \\ & \cdot \delta\{\theta_M - R^{-1}\theta_L = -\sigma \cap \omega_M(t) < R^{-1}\omega_L(t)\}. \end{aligned} \quad (1n)$$

The state equations (1) describe the motor and load shaft velocities and positions and are depicted in block diagram form in Fig. 1. Both shafts include friction torques described in (1e) and (1f). The friction models are comprised of viscous, coulomb and static terms. The gear torque described in (1g) acts on the output shaft and the reflected gear torque reacts on the motor shaft. The expression for gear torque has several terms. The indicator function used in (1h)–(1l) equals “1” when the bracketed condition is true and “0” when false. The delta function used in (1m)–(1n) introduces an impulse when the bracketed condition is true, causing acceleration impulses and velocity step changes. All achievable state conditions are covered by these mutually exclusive terms.

The first term (1h) indicates the gear torque is zero when the gear is in the backlash region. The system is fourth order, and the shafts are completely uncoupled.

The second and third terms (1i)–(1j) give the transmitted torque when the gears are engaged and moving so that the friction is coulomb and viscous. In this condition, the two shafts act as one lumped shaft with the positions and velocities algebraically fixed reducing the system to second order. The simplified state relations are given by (2), where (2a) comes

from using (1i) or (1j) to eliminate  $\theta_L$  and its derivatives from (1c) and using (1c) to eliminate  $\tau_G$  from (1a)

$$(J_M + R^2 J_L) \frac{d\omega_M}{dt} = -\tau_{FM} - R\tau_{FL} + \tau_M \quad (2a)$$

$$\frac{d\theta_M}{dt} = \omega_M \quad (2b)$$

$$\omega_L = R\omega_M \quad (2c)$$

$$\theta_L = R(\theta_M \pm \sigma) \quad (- = \text{positive}, \\ + = \text{negative gear face}). \quad (2d)$$

The reduced system also allows (1a) and (1c) to be solved for  $\tau_G$  (by eliminating the velocity derivatives) giving the expressions found in the max and min arguments. Taking the max and min with zero indicates that torque can only be transmitted in the direction the gear is engaged. If the  $\tau_G$  expression derived from (1a) and (1c) reverses sign,  $\tau_G$  becomes zero and the gears disengage.

The fourth and fifth terms (1k)–(1l) give the torque transmitted when the gears are engaged but not moving, so that the static friction model is used. In this case, the gear reflects to the load shaft that portion of the motor torque—in the direction that the gear is engaged—not canceled by motor shaft static friction.

The sixth and seventh terms (1m)–(1n) introduce torque impulses at the instant when the gears reach the boundaries of the backlash region with sufficient relative velocity to collide. The collisions are inelastic. Thus, the conservation of angular momentum during the collision in (3a) and the condition of zero relative velocity after the collision in (3b) are used to derive the shaft velocities after the collision in terms of the shaft velocities before the collision (3c). Taking the limit of  $J\Delta\omega/\Delta t$  as  $\Delta t \rightarrow 0$  yields the torque impulse in (3d)

$$(J_M + R^2 J_L)\omega_L(t^+) = J_M R\omega_M(t^-) + R^2 J_L\omega_L(t^-) \quad (3a)$$

$$\omega_L(t^+) = R\omega_M(t^+) \quad (3b)$$

$$= (J_M + R^2 J_L)^{-1}(J_M R\omega_M(t^-) + R^2 J_L\omega_L(t^-)) \quad (3c)$$

$$\tau_G(t) = \lim_{\Delta t \rightarrow 0} -R^{-1} J_M \frac{\omega_M(t^+) - \omega_M(t^-)}{\Delta t} \\ = \lim_{\Delta t \rightarrow 0} J_L \frac{\omega_L(t^+) - \omega_L(t^-)}{\Delta t} \\ = J_M J_L (J_M + R^2 J_L)^{-1} (R\omega_M(t^-) - \omega_L(t^-)) \delta(t). \quad (3d)$$

### III. THE CONTROLLER

The nonlinear, state space controller is described by (4) and pictured in Fig. 2. In (4), \* denotes desired, commanded or reference,  $\hat{\cdot}$  denotes estimated,  $\tilde{\cdot}$  denotes error (the desired minus the actual value) and  $\hat{\cdot}$  denotes estimation error (the actual minus the estimated)

$$\tau_G^* := (\hat{J}_L \alpha_L^* + \hat{\tau}_{FL}) + B_{AL}(\omega_L^* - \hat{\omega}_L) + K_{AL}(\theta_L^* - \hat{\theta}_L) \quad (4a)$$

$$\tau_M := \hat{J}_M \alpha_M^* + \hat{\tau}_{FM} + B_{AM}(\omega_M^* - \hat{\omega}_M) \\ + K_{AM}(\theta_M^* - \hat{\theta}_M) + R\tau_G^* \\ \cdot 1\{\theta_R^* = \sigma \cap \tau_G^* > 0\} \cup 1\{\theta_R^* = -\sigma \cap \tau_G^* < 0\} \quad (4b)$$

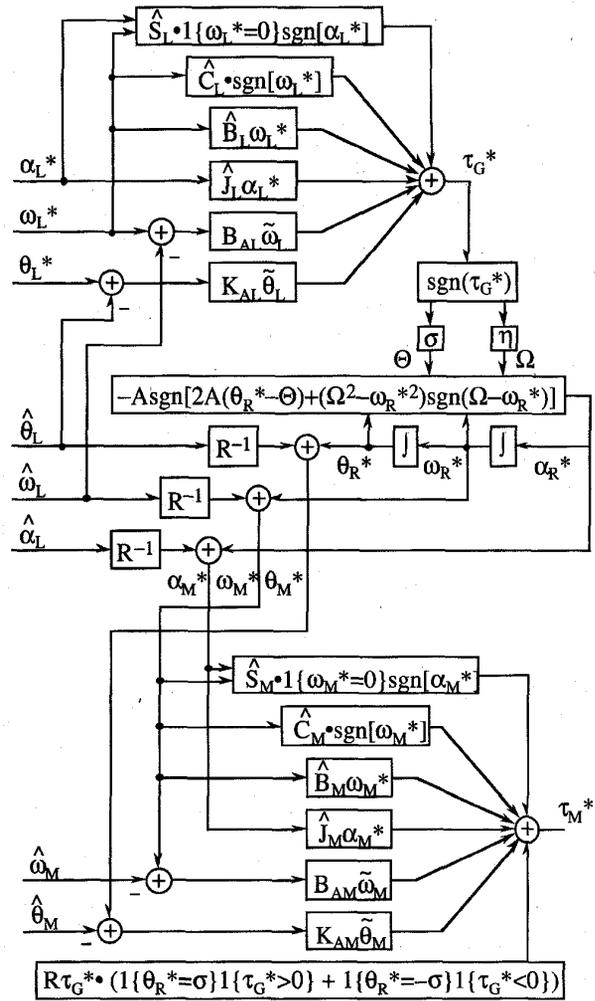


Fig. 2. The nonlinear state space controller.

$$\alpha_M^* := R^{-1} \hat{\alpha}_L + \alpha_R^* \quad (4c)$$

$$\omega_M^* := R^{-1} \hat{\omega}_L + \omega_R^* \quad (4d)$$

$$\theta_M^* := R^{-1} \hat{\theta}_L + \theta_R^* \quad (4e)$$

$$\alpha_R^* := A \cdot [1\{\theta_R^* < SC\} - 1\{\theta_R^* > SC\}] \quad (4f)$$

$$\omega_R^* := \omega_R^*(0) + \int_0^t \alpha_R^*(t) dt \quad (4g)$$

$$\theta_R^* := \theta_R^*(0) + \int_0^t \omega_R^*(t) dt \quad (4h)$$

$$\Theta := \sigma \cdot \text{sgn}(\tau_G^*) \quad \Omega := 0 \cdot \text{sgn}(\tau_G^*) \quad (4i)$$

$$SC := \Theta + (2A)^{-1}(\Omega^2 - \omega_R^{*2}) \cdot [1\{\omega_R^* \leq \Omega\} - 1\{\omega_R^* \geq \Omega\}]. \quad (4j)$$

Since relative motion of the shafts facilitates representing backlash,  $\theta_R := \theta_M - R^{-1}\theta_L$  is used to simplify the equations. Only state variable estimates are used to avoid specifying *a priori* which variables are measured and which are estimated using desired or observed values.

The controller's objective is to make the load shaft follow a desired position trajectory by generating the appropriate motor

shaft torque command and reference trajectory. To do this, the gear torque needed to cause the load shaft to follow the desired trajectory is calculated. Then a motor shaft reference trajectory is generated that uses time-optimal control to guide the motor shaft to (and then hold the shaft at) the correct backlash boundary, making it possible to apply the desired gear torque. Then the motor torque is computed that decouples the motor shaft dynamics, provides state error feedback and, if the gear is engaged, supplies the desired gear torque.

The desired load shaft trajectory segments should be twice differentiable so that the acceleration is continuous. This permits the computation of a corresponding continuous desired transmitted gear torque,  $\tau_G^*$ . This desired torque (4a) provides inertia and friction decoupling and position and velocity error feedback for the load shaft using the model found in [4]. With this definition of  $\tau_G^*$ ,  $\tau_G$  can be eliminated from the load shaft differential equation (1c) and terms grouped to produce (5), which describes the load shaft dynamics in terms of state and parameter estimate errors. This demonstrates the dynamic correctness of  $\tau_G^*$  since the state error differential equation is driven only by parameter estimate errors and errors in the production of  $\tau_G^*$ . When the gears are not engaged,  $\tau_G = 0$  and  $\tilde{\tau}_G = \tau_G^*$ . The gear torque error is then 100%, motivating the use of time-optimal control to minimize the time that this occurs

$$\begin{aligned} J_L \alpha_L &= -\tau_{FL} + \tau_G \quad \text{and} \quad \tau_G = \tau_G^* - \tilde{\tau}_G \Rightarrow \\ 0 &= \hat{J}_L \tilde{\alpha}_L + B_{AL} \tilde{\omega}_L + K_{AL} \tilde{\theta}_L \\ &\quad + B_{AL} \tilde{\omega}_L + K_{AL} \tilde{\theta}_L - \tilde{J}_L \alpha_L - \tilde{\tau}_{FL} - \tilde{\tau}_G. \end{aligned} \quad (5)$$

The motor torque  $\tau_M$  in (4b) decouples motor shaft inertia and friction and adds velocity and position error feedback. If the gears are estimated to be engaged in the correct direction, the desired reflected transmitted torque  $R\tau_G^*$  is also added. The load shaft decoupling terms are built into  $R\tau_G^*$ . Solving (1a) and (4b) and combining terms produces (6a) which is the motor shaft analog to (5). The choice of  $\alpha_M^*$ ,  $\omega_M^*$ , and  $\theta_M^*$  in (4c)–(4e) causes the state error to be the negative of the estimate of the state error allowing (6a) to be simplified to (6b)

$$\begin{aligned} 0 &= \hat{J}_M \tilde{\alpha}_M + B_{AM} \tilde{\omega}_M + K_{AL} \tilde{\theta}_M \\ &\quad + B_{AM} \tilde{\omega}_M + K_{AM} \tilde{\theta}_M - \tilde{J}_M \alpha_M - \tilde{\tau}_{FM} - \tilde{\tau}_G \quad (6a) \\ &= \hat{J}_M \tilde{\alpha}_M - \tilde{J}_M \alpha_M - \tilde{\tau}_{FM} - \tilde{\tau}_G. \end{aligned} \quad (6b)$$

When the gears are not engaged, these two equations are independent, but when the gears are engaged the motor shaft states can be expressed algebraically in terms of the load shaft states. Using this and eliminating  $\tilde{\tau}_G$  from (5) using (6b) yields the combined shaft error dynamics (7)

$$\begin{aligned} 0 &= \hat{J}_L \tilde{\alpha}_L + B_{AL} \tilde{\omega}_L + K_{AL} \tilde{\theta}_L + R^{-2} \hat{J}_M \tilde{\alpha}_L + B_{AL} \tilde{\omega}_L \\ &\quad + K_{AL} \tilde{\theta}_L - (\hat{J}_L + R^{-2} \hat{J}_M) \alpha_L - (\tilde{\tau}_{FL} + R^{-1} \tilde{\tau}_{FL}). \end{aligned} \quad (7)$$

The motor shaft reference trajectory is comprised of acceleration (4c), velocity (4d), and position (4e). Each has a first term that matches the trajectory to the reflected load trajectory and a second term that determines the relative

trajectory. The relative trajectory is described by (4f)–(4j). The desired relative acceleration  $\alpha_R^*$  of (4f) calls for a constant positive acceleration  $A$  when  $\theta_R^*$  is less than SC (the switching condition/curve described below) given in (4j) and a constant negative acceleration  $-A$  when  $\theta_R^*$  is greater than SC. Only on the switching curve is  $\alpha_R^*$  zero, and only at the one stable point  $(\omega_R^*, \theta_R^*) = (0, \Theta)$  can  $\alpha_R^* = 0$  for longer than an instant. The desired relative velocity  $\omega_R^*$  of (4g) is the integral of  $\alpha_R^*$  and is linear and continuous with respect to time. The slopes of the segments are either  $A$ ,  $-A$ , or  $0$ . The desired relative position  $\theta_R^*$  of (4h) is the integral of  $\omega_R^*$  and is parabolic and continuous with respect to time with the curvature of the segments being either  $A$ ,  $-A$ , or  $0$ .

The switching condition/curve is the solution to minimizing the time required to reach a given  $\theta_R^*(t_f) = \Theta$  with  $\omega_R^*(t_f) = \Omega = 0$  subject to  $\alpha_R^* \in [-A, +A]$ . By setting  $\theta_R^*(t_f) = \Theta = \sigma \cdot \text{sgn}(\tau_G^*)$  in (4i), the final relative position is the middle of the backlash region if  $\tau_G^* = 0$  and the proper boundary of the backlash region if  $\tau_G^* \neq 0$ . Thus, the gears are engaged quickly and smoothly with zero relative velocity. Torque can then be transmitted to the load shaft and  $R\tau_G^*$  is added to  $\tau_M^*$ .

In lieu of a formal derivation, a simple argument can be used to justify the result. A calculus of variations proof is found in [5]. Suppose  $\theta_R^* < \Theta$ . To minimize the time, the greatest average velocity must be achieved. Working backward, given  $\omega_R^*$  there is a minimum distance  $\Theta - \theta_R^*$  such that the maximum deceleration  $\alpha_R^* = -A$  will cause  $\omega_R^*$  to just reach zero as  $\theta_R^*$  reaches  $\Theta$ . This is the switching condition because prior to this, the maximum acceleration  $\alpha_R^* = A$  is used to increase and thus maximize the velocity.

The velocity and position equations for a constant acceleration  $\alpha$  are given by (8a) and (9a). The time required to accelerate from  $\omega_R^*(t_1)$  to  $\omega_R^*(t_2)$  is determined by (8b). Then  $t_2 - t_1$  is eliminated from (9a) to give the position trajectory directly in terms of the velocity in (9b). Setting  $t_2 = t_f$  (the final or arrival time) and  $t_1 = t_s$  (the switching time) in (9b) produces (10a), the switching curve in terms of  $\alpha$ . Since  $\alpha = -A$  for  $\omega_R^*(t_s) \geq \Omega$  and  $\alpha = +A$  for  $\omega_R^*(t_s) \leq \Omega$ ,  $\alpha = -A \cdot [1\{\omega_R^* \leq \Omega\} - 1\{\omega_R^* \geq \Omega\}]$  giving the final expression for the switching curve in (10b). Although this controller uses  $\Omega = 0$ ,  $\Omega$  has been retained in the switching curve expression to make it possible to set a small impact velocity to help overcome static friction upon engaging. The most common relative trajectory is to move from one boundary of the backlash region to the other, starting and ending with zero relative velocity and without midcourse reversals. The trajectories for  $\alpha_R^*(t)$ ,  $\omega_R^*(t)$ , and  $\theta_R^*(t)$  for the move from the negative to the positive boundary are pictured in Fig. 3

$$\omega(t_2) = \omega(t_1) + \alpha(t_2 - t_1) \quad (8a)$$

$$\Rightarrow (t_2 - t_1) = \alpha^{-1}(\omega(t_2) - \omega(t_1)) \quad (8b)$$

$$\theta(t_2) = \theta(t_1) + \omega(t_1)(t_2 - t_1) + \alpha(t_2 - t_1)^2/2 \quad (9a)$$

$$\Rightarrow \theta(t_2) - \theta(t_1) = (2\alpha)^{-1}(\omega(t_2)^2 - \omega(t_1)^2) \quad (9b)$$

$$\Rightarrow \text{SC} : \Theta - \theta(t_s) = (2\alpha)^{-1}(\Omega^2 - \omega(t_s)^2) \quad (10a)$$

$$\begin{aligned} \Rightarrow \text{SC} : \theta_R^* - \Theta &= (2A)^{-1}(\Omega^2 - \omega_R^{*2}) \\ &\cdot [1\{\omega_R^* \leq \Omega\} - 1\{\omega_R^* \geq \Omega\}]. \end{aligned} \quad (10b)$$

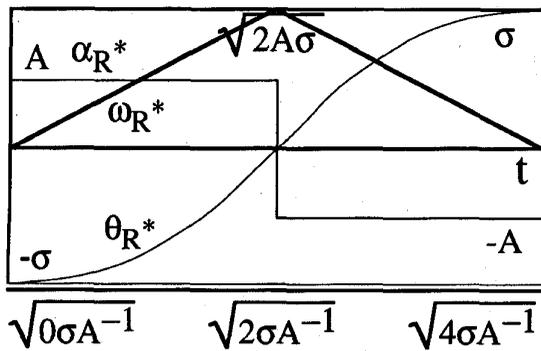


Fig. 3. Acceleration, velocity, and position trajectories in the backlash region.

The relations in (9b) and (10b) provide the basis for a phase plane description shown in Fig. 4.

Positive constant acceleration produces a parabolic trajectory symmetric about the zero velocity line with the curvature in the direction of increasing position and with the evolution in the direction of increasing velocity. Negative constant acceleration produces a parabolic trajectory symmetric about the zero velocity line with the curvature in the direction of decreasing position and with the evolution in the direction of decreasing velocity. The switching curves for the backlash boundaries and zero relative velocity are shown in Fig. 4(a). To the left of the switching curve  $\alpha_R^* = A$ , to the right  $\alpha_R^* = -A$ . Thus,  $\omega_R^*$  and  $\theta_R^*$  follow the parabolic trajectory that evolves into the switching curve, then follow this curve to the desired boundary (or middle) of the backlash region. Fig. 4(b) shows the trajectory for a move from the origin to the negative backlash boundary and then to the positive backlash boundary.

#### IV. NEURAL NETWORK BACKGROUND

Artificial neural networks are interconnections of artificial neurons. The two basic neuron models discussed in [6] are the  $\Sigma$ , the weighted sum of inputs given by (11a) and the  $\Sigma\Pi$ , the weighted sum of products of inputs given by (11b), where  $z$  is the output (or activation) and  $\mathbf{u}$  is the  $n$ -component input vector. Both types of neurons are comprised of simple operations that are readily implemented in VLSI with propagation times in the tens of nanoseconds [7]

$$z = \psi(s) \quad s = w_{\text{bias}} + \sum_{i \in I} w_i u_i \quad (11a)$$

$$I = \{1, \dots, n\}$$

$$z = \psi(s) \quad s = w_{\text{bias}} + \sum_{j \in K} w_j p_j$$

$$p_j = \prod_{i \in I} (1 + (u_i - 1) \cdot 1\{\text{ith digit of } j = 1\})$$

$$K \subset 2^n = \{\text{all } n\text{-digit binary numbers}\}. \quad (11b)$$

The  $\Sigma$ -neuron forms a biased weighted sum ( $s$ ) of the input components and then applies a nonlinear “squashing” function  $\psi$ . The  $\Sigma\Pi$ -neuron forms a biased weighted sum of products ( $p_j$ ) of the input components. Creating all  $2^n$  products is rarely

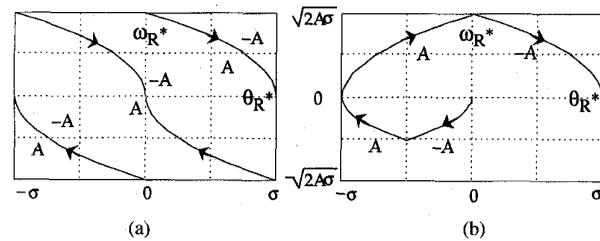


Fig. 4. Phase plane: (a) switching curves and (b) velocity/position trajectory.

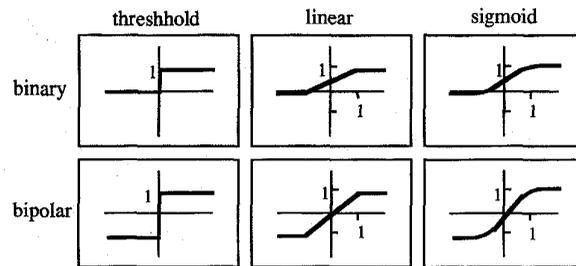


Fig. 5. Possible neuron models.

feasible, so only a subset  $K$  (such as all products with two or fewer terms) is considered. The  $\Sigma$ -neuron is a degenerate  $\Sigma\Pi$ -neuron with only one term products. Product terms with higher powers of a particular  $u_i$  are created by supplying  $u_i$  on as many input lines as the highest desired power.

The squashing function  $\psi$  can be any function that monotonically increases from either  $-1$  (bipolar) or  $0$  (positive) to  $1$ .  $\Psi$  is the “diagonal” operator that takes a vector input and applies  $\psi$  to each component. The most common functions are the sigmoid, the linear, and the threshold pictured in Fig. 5. (Subscripts  $B, P, S, L$ , and  $T$  denote bipolar, positive, sigmoid, linear, and thresholding.) Bipolar and positive units can be interchanged by applying the appropriate bias and scaling factor to the output. Sigmoidal and linear units function as threshold units if the input gains are large. The use of different squashing functions is a notational convenience that facilitates interpreting the network. Simple transformations allow implementing the whole network with a single type of linear or sigmoidal unit.

Neurons are generally arranged in layered feedforward networks such as the two hidden layer network of Fig. 6. In this configuration, the inputs to a neuron are the outputs of the neurons in the previous layer. The output of the neuron then feeds forward to the neurons in the next layer. A key neural network property is that any nonlinear function with bounded input and outputs can be approximated arbitrarily well by a network with at least one hidden layer, enough neurons and correct weights [8].

The usual procedure for finding (training) the weights is an adaptive gradient-descent least squared-error procedure called backpropagation [9]. This process is initialized with random weights which ignore existing knowledge of the system. When the error is linear in the weights, gradient-descent converges quickly and reliably to the best possible solution. However,

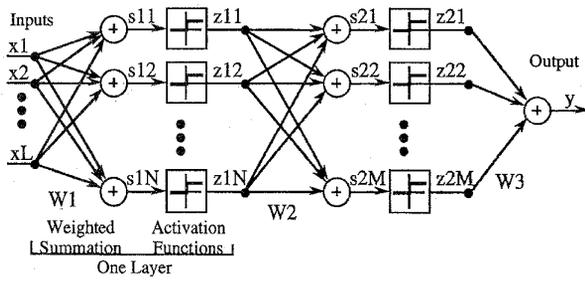


Fig. 6. A neural network with two hidden layers.

in the nonlinear case, gradient-descent may converge slowly, not at all, or to a poor solution. For neural networks, only the output layer has the linear in error property; thus, output layer adaptation works well, but hidden layer adaptation may be slow or fail. To compound matters, the number of neurons needed is found through trial and error, requiring that the procedure be repeated many times.

#### V. NEURON OPERATIONS—NETWORK CONSTRUCTION

The direct construction of the backlash neural controller is based on a building block approach that dedicates a neuron or neurons to the realization of each individual controller function. This physical model-based approach is similar in concept to the synthesis of analog computers as discussed in [10]. Many desired control functions, including linear feedback, bang-bang and logical AND and OR, have extremely simple neuron realizations. More complicated functions encountered in nonlinear decoupling can be approximated using a piecewise reconstruction procedure.

Affine linear operations (linear operations with a bias) are performed by scaling the sum into the linear region of the squashing function as in (11) where  $k$  is the scaling factor. This assumes a linear or sigmoidal bipolar unit

$$\begin{aligned} z &= \psi_{(L \text{ or } S)P} \left( k \cdot \left( w_{\text{bias}} + \sum_{i \in I} w_i u_i \right) \right) \\ &= k \cdot \left( w_{\text{bias}} + \sum_{i \in I} w_i u_i \right). \end{aligned} \quad (11)$$

Therefore, state error feedback and inertia and viscous friction decoupling can be performed by a single neuron. Symmetric bang-bang control is simply a scaled signum function of an input. This is the definition of the bipolar threshold neuron as indicated in (12a). The on-off bang-bang control is the scaled threshold function, which is by definition the positive threshold neuron in (12b)

$$\begin{aligned} z &= k \cdot \psi_{TB}(u) = k \cdot \text{sgn}(u) \\ &= k \cdot [1\{u > 0\} - 1\{u < 0\}] \end{aligned} \quad (12a)$$

$$z = k \cdot \psi_{TP}(u) = k \cdot 1\{u > 0\}. \quad (12b)$$

Technically, the scaling factors  $k$  are not on the neuron output but on the input to all the neurons it feeds.

Neurons can perform the logic operations AND and OR. The  $n$ -input AND and OR neurons are given by (13a) and (13b)

$$\text{AND}(u) = \Psi_{TP} \left( -n + 0.5 + \sum_{i=1}^n u_i \right) \quad (13a)$$

$$\text{OR}(u) = \Psi_{TP} \left( -0.5 + \sum_{i=1}^n u_i \right). \quad (13b)$$

Both neurons start by summing the  $n$  binary inputs. Choosing the bias to be  $-n + 0.5$  means that all  $n$  inputs must be 1 for the sum to be positive and trigger the threshold producing the AND. Choosing the bias to be  $-0.5$  means that at least one input must be 1 to trigger the threshold producing the OR.

More complicated functions can be approximated by using a three-layer network with positive threshold neurons in the two hidden layers and bipolar linear neurons in the output layer. Each first-layer neuron indicates whether the input is in a particular half space. Each second layer neuron AND's designated first-layer neurons, thus indicating whether the input is in the intersection of the designated half spaces. If the input is in the intersection of the designated half spaces, the neuron output is one and the corresponding output weight is added to the linear output; if the input is not in the intersection, the neuron output is zero, contributing nothing to the output. Thus, the input space can be carved into convex regions over which the variation of the desired output is sufficiently small. The regions can then be constructed as the intersection of half spaces and the average desired output over the region assigned to the corresponding output weight. If continuous rather than thresholding neurons are used in defining the half planes, this procedure creates fuzzy sets and, hence, a fuzzy set representation of the function.

#### VI. THE NEURAL NETWORK CONTROLLER

The neural network controller is described by (14a-i) and (15a-f). Equation (14) describes the computation of the desired torque from the desired trajectories ( $\tau_G^*$  from  $\alpha_L^*$ ,  $\omega_L^*$  and  $\theta_L^*$  and  $\tau_M$  from  $\alpha_L^*$ ,  $\omega_L^*$  and  $\theta_L^*$ ). Equation (15) describes the computation of the desired trajectory. As seen by comparing (4a) and (4b) and the controller, in Fig. 2, the equations for  $\tau_G^*$  and  $\tau_M$  are identical except for the addition of

$$R^{-1} \tau_G^* \cdot 1\{[\theta_R^* = \sigma \cap \tau_G^* > 0] \cup [\theta_R^* = -\sigma \cap \tau_G^* < 0]\}$$

to the  $\tau_M$  sum and the use of the appropriate shaft parameters. Therefore, the torque equations are shown only once with a dummy shaft identifier  $S$  which equals  $L$  for load or  $M$  for shaft. An indicator function for  $S = M$  adds the gear torque term to  $\tau_M$ .

$$z1 = \Psi_{TP}(s1) \quad s1 = \begin{pmatrix} 1 & 1 & -1 \\ 0 & -1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \\ -1 & -1 & 1 \\ 0 & -1 & -1 \\ -1 & -1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_S^* T \\ \omega_S^* \\ \Omega \end{pmatrix} \quad (14a)$$

$$z2 = \Psi_{TP}(s2)$$

$$s2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z1 \\ 1.5 \end{pmatrix} \quad (14b)$$

$$\tau_{FFS} = (+\hat{S}_S, +\hat{C}_S, -\hat{C}_S, -\hat{S}_S, +\hat{B}_S, +\hat{J}_S) \begin{pmatrix} \omega_S^* \\ \alpha_S^* \end{pmatrix} \quad (14c)$$

$$\tau_{FBS} = (+B_{AS}, -B_{AS}, +K_{AS}, -K_{AS}) \begin{pmatrix} \omega_S^* \\ \theta_S^* \\ \theta_S \end{pmatrix} \quad (14d)$$

$$z3 = \Psi_{TP}(s3) \quad s3 = \begin{pmatrix} 1 & 0 & -\sigma \\ 0 & 1 & 0 \\ -1 & 0 & -\sigma \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \theta_R^* \\ \tau_G^* \\ 1 \end{pmatrix} \quad (14e)$$

$$z4 = \Psi_{TP}(s4) \quad s4 = \begin{pmatrix} 1 & 1 & 0 & 0 & -1.5 \\ 0 & 0 & 1 & 1 & -1.5 \end{pmatrix} z3 \quad (14f)$$

$$z5 = \Psi_{TP}(s5) \quad s5 = (1 \quad 1 \quad -0.5) z4 \quad (14g)$$

$$\tau_G^* = \tau_{FFS} + \tau_{FBS} \quad \text{with } s = L \quad (14h)$$

$$\tau_M = \tau_{FFS} + \tau_{FBS} + R\tau_G^* \cdot z5 \quad \text{with } S = M \quad (14i)$$

$$z6 = \Psi_{TB}(\tau_G^*) \quad (15a)$$

$$z7 = \Psi_{LB}(s7) \quad s7 = \begin{pmatrix} 1 & 0 & -\sigma \\ 0 & 1 & \eta \\ 0 & 1 & -\eta \\ 0 & G & -G\eta \end{pmatrix} \begin{pmatrix} \theta_R^* \\ \omega_R^* \\ z6 \end{pmatrix} \quad (15b)$$

$$z8 = \Psi_{TB}(s8) \quad s8 = \left(1 - \frac{1}{2A}\right) p8$$

$$p8 = \begin{pmatrix} z7_1 \\ z7_2 \cdot z7_3 \cdot z7_4 \end{pmatrix} \quad (15c)$$

$$\alpha_M^* = R^{-1}\hat{\alpha}_L + \alpha_R^* \quad \alpha_R^* = -A \cdot z8 \quad (15d)$$

$$\omega_M^* = R^{-1}\hat{\omega}_L + \omega_R^* \quad \omega_R^* = \omega_R^*(0) + \int_0^t \alpha_R^*(t)dt \quad (15e)$$

$$\theta_M^* = R^{-1}\hat{\theta}_L + \theta_R^* \theta_R^* = \theta_R^*(0) + \int_0^t \omega_R^*(t)dt. \quad (15f)$$

Equations (14a)–(14d) and (14h)–(14i) are essentially the neural network position controller with friction decoupling equations demonstrated in [11]. Two slight changes have been made. To accommodate the continuous time formulation of the controller equations used here,  $\omega^*(t+1)$  was replaced with  $\omega^*(t) + \alpha^*(t)T$ . Also, the gear torque term had to be added to the  $\tau_M$  sum. *This illustrates the building block principle by using two single-shaft networks from [11] to build a two-shaft controller by connecting the networks with the correct torque coupling term.*

In (14a), the first and second  $z1$  components are the indicators for the two conditions for including positive static friction decoupling. The third and fourth  $z1$  components indicate the conditions for including positive coulomb friction decoupling. The fifth and sixth  $z1$  components indicate the symmetric conditions for negative coulomb friction decoupling. The seventh and eighth  $z1$  components indicate the symmetric conditions for the negative static friction decoupling. In (14b),

the four components of  $z2$  are the AND's of these pairs of  $z1$  components. ANDing set indicators is equivalent to indicating the intersection of the sets as discussed in the piecewise function approximation procedure in the previous section. Thus, the  $z2$  components indicate the use of  $\hat{S}, \hat{C}, -\hat{C}$  and  $-\hat{S}$ , respectively. In (14c), the complete feedforward torque decoupling is computed by appending  $w^*$  and  $\alpha^*$  to  $z2$  and setting the weights to the corresponding parameter estimates. In (14d), the state error feedback torque is computed.

In (14e), the conditions required to form the  $R\tau_G^*$  indicator function for the  $\tau_M$  sum are created. The first  $z3$  component indicates the gear is positively engaged, and the second  $z3$  component indicates  $\tau_G^*$  is positive. The third and fourth components indicate the gear is negatively engaged and  $\tau_G^*$  is negative. In (14f), the first  $z4$  component ANDs the first two conditions, and the second  $z4$  component ANDs the second two conditions. In (16g),  $z5$  ORs the two components of  $z4$  producing the desired indicator.

In (14h) and (14i), the torque commands are produced by summing the feedforward decoupling and feedback terms. For the motor shaft, the gear torque decoupling is also included using  $z5$  to switch the term appropriately. This  $R\tau_G^*$  multiplied by its indicator function followed by the summation has the form of  $\Sigma\Pi$ -neuron.

The relative trajectory and the motor trajectory are formed in (15). In (15a),  $z6$  serves as the  $\text{sgn}(\tau_G^*)$  term used in determining  $\Theta$  and  $\Omega$ . In (15b), the first  $z7$  component is  $\theta_R^* - \Theta$ . The second and third  $z7$  components are  $\omega_R^* + \Omega$  and  $\omega_R^* - \Omega$ . The fourth  $z7$  component is  $\text{sgn}(\omega_R^* - \Omega)$ , where  $G$  is assumed to be a large transforming, the linear to a threshold unit. In (15c), the switching condition is formed with a  $\Sigma\Pi$ -neuron. The first  $p8$  product term is  $\theta_R^* - \Theta$ . The second  $p8$  product term forms  $(\Omega^2 - \omega_R^{*2})\text{sgn}(\omega_R^* - \Omega)$  as the product of the second, third, and fourth components of  $z7$ . The second product term is scaled by  $(2A)^{-1}$  and the scaled products are summed, creating the switching condition. The bipolar threshold activation function performs the switch making  $z8$  the indicator for using  $+A$  or  $-A$  relative acceleration. The relative acceleration is so set in (15d), with (15e) and (15f) giving the corresponding integral relations for the relative velocity and position.

## VII. RESULTS

The experiment was performed using a commercial dc motor controlled by a PC-AT with a Burr-Brown data acquisition board operating at a 200-Hz sampling rate. Encoders were mounted on both shafts, but only the motor shaft encoder was used for parameter estimation and control. When the relative trajectory reference indicated the gears were engaged, the algebraic relation between the load and motor shaft was used. When not engaged, the desired load shaft trajectory was used as an estimate. The parameter estimates and gains used are as follows:

$$\begin{aligned} \hat{S}_L &= 0.025 & \hat{S}_M &= 0.025 \text{ N} \cdot \text{m} \\ \hat{C}_L &= 0.009 & \hat{C}_M &= 0.006 \text{ N} \cdot \text{m} \\ \hat{B}_L &= 0.000 & \hat{B}_M &= 0.12 \times 10^{-3} \text{ N} \cdot \text{m}/(\text{rad/s}) \end{aligned}$$

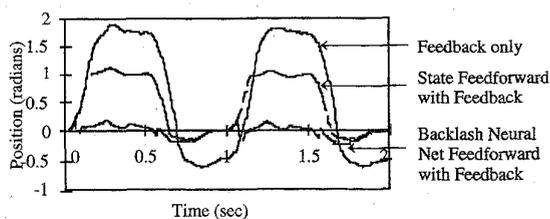


Fig. 7. Error comparison for 1-Hz parabolic trajectory.

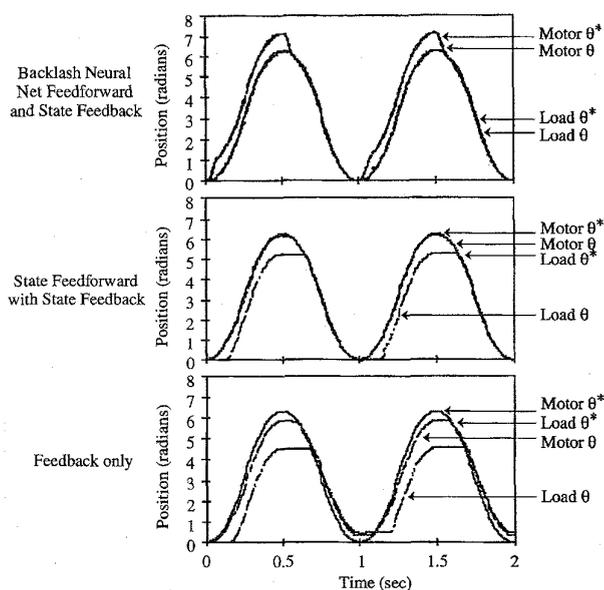


Fig. 8. Motor and load/reference and actual trajectories.

$$\begin{aligned} \hat{J}_L &= 0.10 \times 10^{-5} & \hat{J}_M &= 0.10 \times 10^{-3} \text{ N} \cdot \text{m}/(\text{rad}/\text{s}^2) \\ B_{AL} &= 6.80 \times 10^{-5} & B_{AM} &= 6.80 \times 10^{-3} \text{ N} \cdot \text{m}/(\text{rad}/\text{s}) \\ K_{AL} &= 0.041 \times 10^{-2} & K_{AM} &= 0.041 \text{ N} \cdot \text{m}/(\text{rad}) \\ R &= 1 & \sigma &= 0.44 \text{ rad} & A &= 75 \text{ rad}/\text{s}^2. \end{aligned}$$

The estimate  $\hat{J}_L$  is not accurate. We only know  $\hat{J}_L < 0.02\hat{J}_M$ . Thus,  $\hat{J}_L$ ,  $B_{AL}$  and  $K_{AL}$  were chosen proportionate to the optimally tuned corresponding motor parameters.

The results for a 1-Hz parabolic load trajectory are shown in Figs. 7 and 8. Comparisons are made between controllers with feedback only, feedback plus feedforward decoupling using lumped shaft parameter estimates, and the backlash controller. Note that for convenience, the motor shaft was referenced to the negative backlash boundary rather than the center. This creates large error on positive movement and smaller error on negative movement, rather than symmetric error.

The results are consistent for the two cases. With feedback only, the motor shaft lags its desired trajectory in addition to the lag and flat regions caused by failing to alter the motor shaft trajectory. Addition of the feedforward decoupling allowed the motor to track its reference well, but still left the characteristic backlash lag and flat zone. The backlash controller supplied the quickest trajectory correction possible

given the current limit of our drive. It left only a short small pulse in the load shaft error while the motor shaft moved to the correct gear face.

In this experiment, the error reductions relative to the feedback-only controller are approximately 50% for the addition of feedforward and 80% to 90% for the backlash controller. However, the percent reductions can vary greatly from system to system, depending on the relative magnitude of the backlash-induced errors to the friction-induced errors and how well the parameters can be estimated. Position dependence of the friction and shaft compliance can also be significant factors that have the potential to mask the benefits of this controller.

## VIII. CONCLUSION

A neural network topology that compensates for backlash and friction in precision, position-controlled mechanisms has been developed and successfully demonstrated. The control algorithm it implements has several unique features:

- It moves the motor shaft to the proper backlash boundary using a time-optimal control subject to acceleration limits.
- It decouples the inertial, friction (viscous, coulomb, and static) and the gear torque.
- It incorporates properly referenced state error feedback. The method of implementing the neural network embodies several novel features.
- Standard linear control is directly implemented by a neuron operating in its linear region. This allows the corresponding weights to be initialized with the gains of an existing controller.
- The bang-bang control that results from time-optimal control subject to hard constraints can be directly implemented by a thresholding neuron once the switching curve is created.
- State conditions for inclusion of friction decoupling, gear torque decoupling, and the optimal relative acceleration were systematically created from  $\Sigma$ -neuron half-spaces and a  $\Sigma\Pi$ -neuron parabola.
- Logical operations were used to create complicated conditions from more basic, single neuron-generated conditions.

## REFERENCES

- [1] F. G. Shinskey, *Process Control Systems*. New York: McGraw-Hill, 1967, pp. 128-130.
- [2] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991, pp. 171-179.
- [3] G. Tao and P. V. Kokotovic, "Adaptive control of systems with backlash," *Tech. Rep. CCEC*, UCSB, 1992.
- [4] C. T. Johnson and R. D. Lorenz, "Experimental identification of friction and its compensation in precise, position controlled mechanisms," in *Proc. IEEE Ind. Applicat. Soc. Annu. Meeting*, Oct. 1991, pp. 1392-1398.
- [5] F. L. Lewis, *Optimal Control*. New York: Wiley, 1986, pp. 252-280.
- [6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, pp. 45-76.
- [7] C. Narathong, University of Wisconsin-Platteville Rep., ECE Dept.
- [8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, 1989.

- [9] D. E. Rumelhart, G. E. Hinton and J. L. McClelland, "A general framework for parallel distributed processing," in *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, pp. 318-364.
- [10] C. L. Johnson, *Analog Computer Techniques*. New York: McGraw-Hill, 1956.
- [11] D. R. Seidl, T. L. Reneking, and R. D. Lorenz, "Use of neural networks to identify and compensate for friction in precision, position controlled mechanisms," in *Proc. IEEE Ind. Applicat. Soc. Annu. Meeting*, Oct. 1992, pp. 1937-1944.



**David R. Seidl** received the B.S. and M.S. degrees in electrical engineering from the University of Wisconsin, Madison, in 1985 and 1990.

He is a Ph.D. student in electrical engineering at the University of Wisconsin as a Rockwell Foundation Fellow. He is currently a development engineer at UNICO, Inc., Franksville, WI, applying optimal, nonlinear, and robust control to drives and motion control systems, and a consultant for QualTim Technologies working on the nondestructive testing of wood products. His other research interests include

neural network and fuzzy logic system identification and control.

Mr. Seidl is a member of the IEEE Industry Applications Society and Control Systems Society.



**Sui-Lun Lam** received the B.S. degree in mechanical engineering from the University of Washington, Seattle, in 1986, the B.S. degree in mathematics from the University of Wisconsin, Madison, in 1988, and the M.S. degree in manufacturing systems engineering from the University of Wisconsin, Madison, in 1995.

He is currently working as an information system manager at Secure Medical Products Inc., White-water, WI.



**Jerry A. Putman** received the B.S. degree in electrical engineering from the University of Nebraska, Lincoln, in 1988.

He served as an officer in the U.S. Army (Signal Corps) from 1988 to 1992. He is currently a research assistant and is studying for the M.S. degree at the University of Wisconsin, Madison. He is a member of the Wisconsin Electric Machines and Power Electronics Consortium. His research areas include fuzzy logic control, expert systems, neural networks, and electrical power conversion.



**Robert D. Lorenz** (S'83-M'84-SM'91) received the B.S., M.S., and Ph.D. degrees from the University of Wisconsin, Madison, in 1969, 1970, and 1984 respectively. In 1969 and 1970, he did his Master's thesis research at the Technical University of Aachen, West Germany.

Since 1984, he has been a member of the faculty of the University of Wisconsin, Madison, where he is Professor of mechanical engineering and of electrical and computer engineering. In this position, he acts as Associate Director of the Wisconsin

Electric Machines and Power Electronics Consortium and as Co-Director of the Advanced Automation and Robotics Consortium. He was a Visiting Research Professor in the Electrical Drives Group of the Catholic University of Leuven, Belgium, and in the Electrical Drives Institute of the Technical University of Aachen, West Germany, in the summers of 1989 and the summers of 1987, 1991, and 1995, respectively. From 1972 to 1982, he was a member of the research staff at Gleason Works, Rochester, NY. He is an active consultant to many organizations. His current research interests include sensor integrated electromagnetic actuator technologies, real time digital signal processing and estimation techniques, and ac drive and high precision machine control technologies.

Dr. Lorenz is a past Chairman of the IEEE Industry Applications Society Industrial Drives Committee, and is a member of the Industrial Automation and Control Committee, the Electrical Machines Committee, and the Industrial Power Converter Committee. He is a Registered Professional Engineer in the States of New York and Wisconsin. He is a member of ASME, ISA, and SPIE.