

NEURAL NETWORK CONTROL OF NONMINIMUM PHASE SYSTEMS BASED ON A NONCAUSAL INVERSE

Andrew H. Register

Georgia Institute of Technology
Georgia Tech Research Institute
400 10th Street, CRB 614
Atlanta, GA 30332

Wayne J. Book

Georgia Institute of Technology
School of Mechanical Engineering
Atlanta, GA 30332

Cecil O. Alford

Georgia Institute of Technology
School of Electrical and Computer Engineering
Atlanta, GA 30332

ABSTRACT: A new approach for feedforward ANN control of nonminimum phase mechanical systems is proposed. A standard backpropagation-of-errors ANN is used to form an inverse model controller which is applied to simulated nonminimum phase systems. Learning in the new approach is based on the convolution between a noncausal impulse response and a desired tip trajectory. Selection of the proper input set, input scaling and the ANN structure are investigated. Once the input and structure are specified, the ANN is trained over a single trajectory. After training, the ANN is used to drive the system in an open-loop configuration. Plots of the system states resulting from the ideal excitation and from ANN excitation are compared. The results obtained by varying both the number of units and the input set are presented. The results demonstrate the effectiveness of the proposed ANN inverse model approach.

I. INTRODUCTION

Although the first study of artificial neural network (ANN) learning by McCulloch and Pitts occurred over 50 years ago, the use of ANNs did not become popular until more recently. Researchers have applied a variety of ANN technologies to the control of mechanical structures and survey papers specifically related to ANN control of rigid mechanical systems may be found in the literature (e.g., [1]). The use of ANN learning control for nonminimum phase systems has also been studied, although less extensively. The studies related to robotics, center on a particular type of nonminimum phase system, a flexible link manipulator.

A three-layer perceptron has been used to control a four-bar mechanism through a flexible coupling[2]. Flexible joint manipulators, however, do not exhibit nonminimum phase behavior, so the results for flexible joint manipulators are not directly applicable to flexible link manipulators. A recurrent, three-layer perceptron has been used to control a scale model of a space-based, flexible manipu-

lator with both joint and link flexibility[3]. The investigation, however, used low-pass filters and small gain constants to prevent excitation of vibration modes. In addition, neither strain nor tip measurements were used as inputs to the ANN controller. Thus, the experiment was an extension of the rigid link case. A multi-layer perceptron has been used for payload identification and gain selection for the control of a single-link flexible manipulator[4]. The ANN is not used as a controller but rather as a pattern classifier. Once the payload is identified and a gain selected, the linear control law does not change until the payload changes. An array of four multi-layer perceptrons has been used to manipulate a simulated flexible plate[5]. Beam shape information was provided to the ANN using a novel gripper design. A multi-layer perceptron has also been used to control the tip-position of a single-link flexible manipulator[6]. The proposed ANN controller had a strong reliance on an external teacher, had an ANN structure of sparse interconnects, and used linear activation functions. No strain or tip measurements were used by the ANN and the results include only limited simulations. Perhaps the most promising research for ANN based flexible manipulator control used a multi-layer perceptron to control the tip position of a single-link flexible manipulator[7]. The investigation used an ANN controller with access to both traditional state measurements and strain measurements. The network consisted of one hidden layer with a surprisingly small number of units. Comparison with a fixed-gain controller showed advantages for the multi-layer perceptron. The research in [7] was shown to have certain limitations related to nonminimum phase characteristics[8]. In these referenced studies, nonminimum phase characteristics of the flexible link systems were not addressed during the design of the ANN controller.

Now consider the block diagram in Fig. 1 where $H(s)$ is the system plant and $H^{-1}(s)$ is the inverse. In order to make the cascaded transfer function of plant and inverse equal to one, the inverse of the plant would need to have zeros where the plant had poles and poles where the plant had zeros. In Fig. 1, the so called 'inverse

model controller' configuration, the inverse is used to condition commands so that the system output will follow the desired input. The inverse model configuration is a convenient way to use ANNs to control a rigid mechanical system[1]. For nonminimum phase systems, however, the plant has right-half-plane (RHP) zeros and the inverse will have RHP poles. If a causal inverse is considered, the inverse is unstable. If the causality condition is relaxed, it is well known that residue calculus can be used to find the noncausal, time-domain response of a system with poles in the RHP[9]. The use of a noncausal inverse to control a flexible manipulator has been reported[10][11][12]. To control such a nonminimum phase system, input torque is calculated by solving the convolution integral between the noncausal impulse response of the system inverse and the desired trajectory. The convolution may be computed indirectly in the frequency domain and transformed to the time domain[10], or the convolution may be computed directly in the time domain[11][12]. ANN methods for determining the inverse for nonminimum phase systems have been proposed. These methods do not appear to have been applied to flexible link mechanical systems.

An inverse model control method based on an adaptive FIR filter has been proposed for the control of nonminimum phase systems[13]. In [13], the proposed use of a so called 'delayed inverse identification process' for nonminimum phase systems was an important contribution. The method of [13] was extended to use an ANN in place of the FIR filter[14]. The extension considered nonlinear systems but did not consider nonminimum phase systems. There has also been some doubt cast over the direct inverse identification process required by the method of [14]. It has been argued that the identification process requires specially formed inputs and there is no guarantee that the correct inverse will be found in problems where the inverse is not one-to-one[18]. There is also no reported evidence that the method of [14] will work when the input and output are non-collocated (an added complexity with flexible link manipulators).

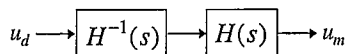


Figure 1, Inverse model controller.

Other ANN based controllers for nonminimum phase systems have been proposed. A conventional, proportional-plus-integral controller with an ANN gain adjustment was proposed to control the nonminimum phase plant displayed by a steam generator[16]. There are no delays in the learning path and by the author's observation, "the effect of the non-minimum phase behavior is still very much pronounced." [16, pg. 64] A partially recurrent, ANN method has been proposed for use in a finite horizon model predictive control framework[17]. In the finite horizon paradigm, the ANN must accurately emulate the forward characteristics of the plant and not the inverse characteristics. Predictive methods do not rely on obtaining an inverse and will not be considered in this study.

There is high confidence that an ANN has the capability to learn the required torque commands because it has been shown that an ANN is capable of representing an arbitrary mapping, $\mathfrak{R}^n \rightarrow \mathfrak{R}^m$, on a compact domain to an arbitrary accuracy[18]. Unfortunately, there

is no analytical method for determining the required inputs, the input scaling, and the network structure (e.g., number of layers, the number of units per layer, etc.) necessary to obtain a particular mapping. Here, the investigation centers on finding an input set, input scale, and ANN structure to allow a conventional, backpropagation-of-errors ANN to learn the required torque function over a single trajectory. The investigation includes a consideration of the noncausal characteristics of the inverse and the amount of delay required.

Two systems will be considered. The first system is the simple nonminimum phase system proposed in [12]. The second system is the model of a flexible link manipulator developed in [8]. In both cases, there are no damping terms included in the system models. Because of the lack of damping, any error in the driving input will cause significant steady state vibration. The absence of significant vibration is an indication that the ANN has learned the required input/output relationship.

II. FEEDFORWARD ANN CONTROLLER

A. Simple System Equations

A simplified system has been proposed to aid in the understanding of problems associated with nonminimum phase systems[12]. The dynamic equation for the two degree-of-freedom system is

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} f(t) \\ 0 \end{bmatrix}, \quad (1)$$

resulting in a transfer function between output, $Y(s)$, and input, $F(s)$, given by,

$$Y(s) = -\frac{1}{3} \frac{s^2 - 1}{s^2(s^2 + 2)} F(s). \quad (2)$$

Eqn. 2, with zeros at $s = \pm 1$, is clearly nonminimum phase. Inverting Eqn. 2 by replacing poles with zeros and zeros with poles results in an inverse that can be used to find the required input, $F_r(s)$. The inverse is given by,

$$F_r(s) = -3 \left(\frac{s^2(s^2 + 2)}{s^2 - 1} \right) \left(\frac{Y_d(s)}{s^2} \right), \quad (3)$$

where $Y_d(s)$ represents the desired acceleration of the tip (i.e., $s^2 Y_d(s)$). The inverse in Eqn. 3 has a RHP pole and is not stable

under causal assumptions. Residue calculus can be used to find a noncausal, stable solution to Eqn. 3. Once the impulse response has been found, the required time domain input may be found by the convolution,

$$f_r(t) = \left(-3u_0(t) + \frac{9}{2}e^{-t}u_{-1}(t) + \frac{9}{2}e^t u_{-1}(-t) \right) * \frac{d^2 y_d(t)}{dt^2}. \quad (4)$$

B. Simple System ANN Input/Output

Before an ANN may be proposed as an inverse model controller, the ability of an ANN to accurately learn the result of the convolution in Eqn. 4 must be demonstrated. To make the results comparable to previous results in the literature, the trajectory for this investigation was chosen to be the bang-bang acceleration from [12] given by,

$$\frac{d^2 y_d(t)}{dt^2} = \begin{cases} 0, & t < 0 \\ +1, & 0 \leq t < 1 \\ -1, & 1 \leq t < 2 \\ 0, & t \geq 2 \end{cases} \quad (5)$$

Plots of required force versus time and the states, $x(t)$ and $y(t)$, for the desired trajectory in Eqn. 5 are shown in Fig. 2. The force curve in Fig. 2 represents the function that must be learned by the ANN.

Fig. 2 shows that the required input force is discontinuous at a finite number of locations. A four-layer ANN (i.e., two hidden layers) has been shown to be capable of approximating functions with a finite number of discontinuities[18]. A four-layer ANN will be used to learn the force relationship.

The most basic model of an ANN as a function approximator is a black-box that accepts one or more inputs and produces one or more outputs. Implied in this model are certain requirements on the mapping between ANN inputs and outputs. Each unique desired output must be derived from at least one distinct input value (e.g., different output values cannot be obtained for the same input value, and multiple input values may map to the same output value). A continuously increasing function throughout the range where the output is changing represents one acceptable class of inputs. Time is a good example of a continuously increasing function. It can be shown that using time as an input produces a good approximation, however, time may not be the best choice.

If the ultimate objective is to follow a desired tip trajectory, the ANN input should include some representation of the desired tip motion. Unfortunately, for nonminimum phase systems, time-scale differences between the desired tip position and the required force do not allow the desired tip motion to be directly used as an input. The desired tip motion can be manipulated, however, to provide an acceptable input.

A variety of transformations on the input may be considered. For the simple system, only one transformation between the desired tip

position and the ANN input is investigated. An additional transformation is investigated later, for use with the flexible link system.

The simple system is time-invariant, allowing for the consideration of an input transformation expressed by,

$$y_{in,ann}(t) = K_a y_d(k_s(t - \tau_a) - \tau) - K_o, \quad (6)$$

where $y_d(\cdot)$ is the desired tip trajectory, τ_a is a time advance constant, k_s is a time stretch constant, K_a is a gain constant, and K_o is an offset constant. The result of applying the transformation in Eqn. 6, using particular values of τ_a , k_s , K_a , and K_o , is shown in Fig. 3.

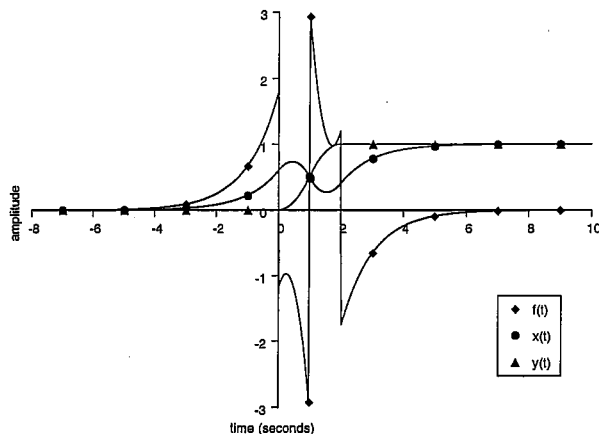


Figure 2, Force and states for the simple system.

Particular values of τ_a and k_s are determined based on the time values where $f_r(t) = \text{constant}$ for all $t < \tau_a$, and $f_r(t) = \text{constant}$ for all $t > \tau_m$. From Fig. 2, the time where the input force first becomes significant corresponds to the value for τ_a and the time where the input force permanently returns to a value near zero corresponds to the value for τ_m . Input scale values, K_a and K_o , affect the learning performance of the ANN and are selected by trial and error.

One consequence of advancing and stretching the desired input function is that nonminimum phase zeros close to the $j\omega$ axis will require a large value for τ_a and left-half-plane zeros close to the $j\omega$ axis will require a large value for τ_m . Large values make it more difficult to find the inverse dynamics because the spread of the plant is large[19]. Thus, large flexible structures with low natural frequencies represent a particularly challenging problem.

Before a particular advance time value may be determined, the meaning of 'significant force' must be addressed. Realistically, a significant value for input force would be determined by the precision of the components in the system. Assuming a precision of 12 bits and

an output range of ± 5 , the smallest possible output value is approximately 2.5×10^{-3} . In this example, a value of 2.5×10^{-3} is obtained at approximately $\tau_a = -6.6$ seconds. Likewise, the force permanently returns to a value within 2.5×10^{-3} of zero at $\tau_m = 8.6$. The resulting stretch factor required to stretch the total motion time over $\tau_m - \tau_a = 15.2$ seconds is 0.13.

For a real system, the existence of τ_a in Eqn. 6 implies that there must be a constant time delay between the desired trajectory and the resulting measured trajectory. For example, if the desired trajectory is specified beginning at $t = 0$, the measured trajectory should not begin before $t = \tau_a$. The delay is necessary because of the way the noncausal function is being approximated. In the simulated results, however, there are no physical barriers to achieving negative time values. In the simulations, the force begins at some negative time and the desired motion begins at time equal to zero.

C. ANN Approximation

A four layer ANN structure with one input and one output was used in an attempt to learn the force function in Fig. 2. A variety of input scale functions and variety of values for τ_a and k_s were examined. The stretch and shift values derived directly from Fig. 2. did not produce acceptable results. The minimum values for τ_a and τ_m had to be extended slightly to -7.5 and 9.5 seconds, respectively, resulting in a value for k_s of 0.118. The input scale function found to produce the best results for this particular desired trajectory is given by,

$$y_{in,ann}(t) = 40y_d(0.118(t + 7.5) - \tau) - 20. \quad (7)$$

The stretched and scaled input function is shown in Fig. 3.

Using the scale function from Eqn. 7, the required number of units in the hidden layer was investigated. Increasing the number of units in the hidden layer, results in a continuous reduction in the error between the ideal force and the ANN generated force, between the simulated tip position and the desired tip position and in the residual vibration. A 1-2-2-1 network (figure not shown, $F_{error,rms}=0.32$, $P_{error,rms}=1.5$, $Vibration_{p-p}=0.36$) was inadequate for both gross slewing motion and vibration. A 1-6-6-1 network (Fig. 4, $F_{error,rms}=0.092$, $P_{error,rms}=0.033$, $Vibration_{p-p}=0.073$) resulted in adequate gross slewing with some residual vibration. Increasing from 1-6-6-1 to 1-8-8-1 (Fig. 5, $F_{error,rms}=0.089$, $P_{error,rms}=0.062$, $Vibration_{p-p}=0.060$) resulted in an 18% reduction in the residual vibration. A 1-12-12-1 network (figure not shown, $F_{error,rms}=0.075$, $P_{error,rms}=0.051$, $Vibration_{p-p}=0.056$) reduced the residual vibration by an additional 7%. This result leads to the observation that the number of units in the network may be selected based on an application's measure of acceptable, residual vibration.

Units in both hidden layers and in the output layer used a hyperbolic tangent activation function. The output layer values were scaled to range between ± 5 . Learning was performed by sampling the

ideal force function, Fig. 2, and randomly presenting the samples to the ANN. Approximately 800 uniformly distributed samples of the input and desired force from -7.5 to 9.5 seconds were used. Extending the sampling beyond these limits caused the initial and final force values to be closer to zero at the expense of a less accurate active region and larger steady state oscillation. A higher sample rate did not affect the results except that the learning times were longer. A lower sample rate did not allow the learned force to exhibit sharp transitions, resulting in a slightly larger steady state oscillation. Non-uniform sampling was also attempted. Sampling the input and the desired force function so that the absolute value of the force change between two adjacent samples equaled a constant drove the solution into a local minimum that did not produce acceptable results. Uniform sampling augmented with non-uniform, force-change samples caused large errors in the initial and final force values after learning.

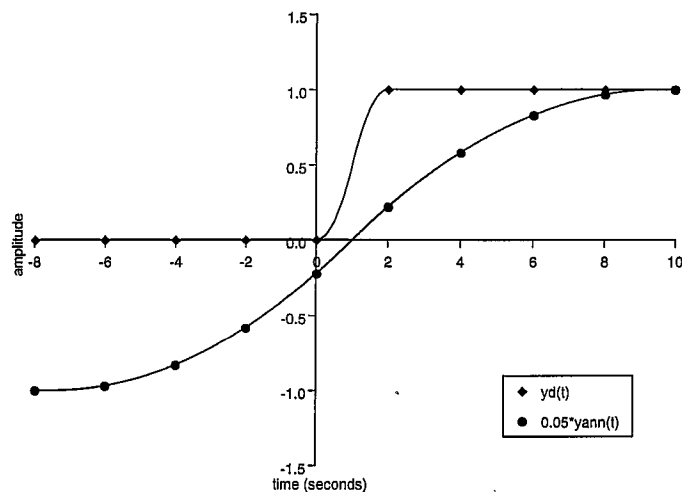


Figure 3, Example of the shift and stretch input transform.

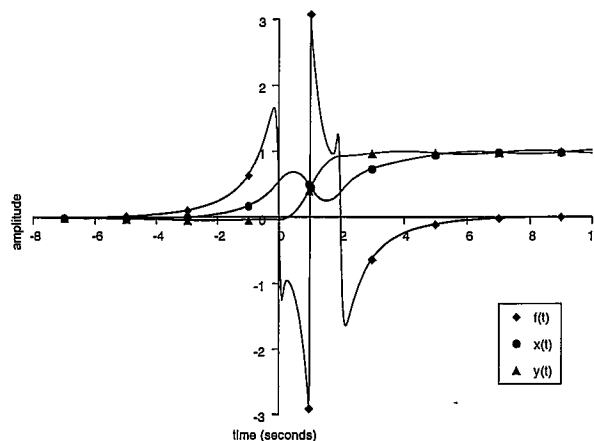


Figure 4, Results for 1-6-6-1 ANN.

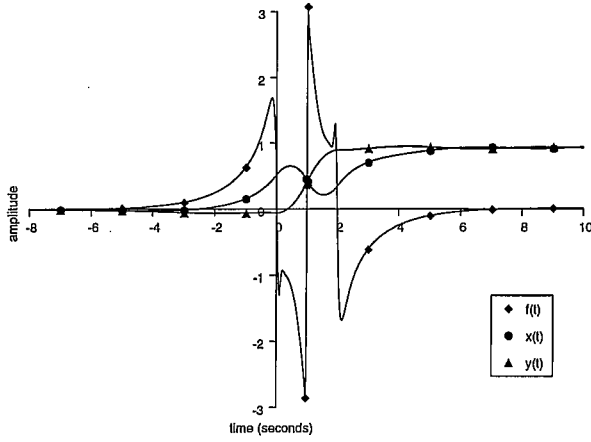


Figure 5, Results for 1-8-8-1 ANN.

The weights were updated after each sample presentation using the standard backpropagation-of-errors method (BackProp). The BackProp learning rate was adjusted downward when learning at the current rate slowed. The initial rate was 0.05 and the final rate was 0.0001. Approximately 7500 random presentations of the sample set were required. Epoch based learning was also attempted. Epoch based learning consistently converged to a local minimum that resulted in an unacceptable system response. In both cases, epoch based and sample based, an inertia term was not added to the BackProp algorithm.

D. Flexible Link System Equations

For flexible link manipulators, an assumed-modes method, coupled with a Lagrangian technique, yields a recursive, closed-form, dynamic solution suitable for control purposes[20]. Systematic application of the Lagrangian, a two mode vibration assumption and some simplifying assumptions, yield the undamped, linearized, inverse dynamic equation[8] used in this investigation. Substitution of the particular physical parameters from [8] ($J_0 = 0.01 \text{ Nms}^2$, $\rho = 2700 \frac{\text{Kg}}{\text{m}^3}$, $w = 0.001 \text{ m}$, $h = 0.02 \text{ m}$, $m_l = 0.05 \text{ Kg}$, and $E = 71.0 \times 10^9 \text{ Pa}$) and a beam length of 1.4 m results in a specific dynamic equation given by

$$\begin{bmatrix} 0.16 & -0.059 & 0.019 \\ -0.059 & 0.24 & 0 \\ 0.019 & 0 & 0.071 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3.66 & 0 \\ 0 & 0 & 12.4 \end{bmatrix} \begin{bmatrix} \theta \\ \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -3.30 \\ -1.15 \end{bmatrix} \tau, \quad (8)$$

where θ is the rigid body angle (not necessarily the hub angle) and ϕ_i is a set of time-dependent generalized coordinates that modulate the amplitude of the beam vibration. Rearranging the dynamic equation yields a transfer function between tip position and torque given by,

$$Y_i(s) = 1.72 \frac{(s+5.5)(s-5.5)(s+19.5)(s-19.5)}{s^2(s^2+17)(s^2+182)} \tau(s). \quad (9)$$

Eqn. 9, with zeros at $s = \pm 5.5$ and $s = \pm 19.5$, is clearly nonminimum phase. By using the same procedure used with the simple system, the impulse response was determined. The required time domain torque may be found by the convolution,

$$\begin{aligned} \tau_r(t) = & (0.58u_0(t) + 15e^{-5.5t}u_{-1}(t) - 9.5e^{-19.5t}u_{-1}(t) \\ & + 1.5e^{5.5t}u_{-1}(-t) - 9.5e^{19.5t}u_{-1}(-t)) * \frac{d^2 y_{i,d}(t)}{dt^2} \end{aligned} \quad (10)$$

E. Flexible Link ANN Input/Output, Bang-Bang Accel.

Using the desired bang-bang acceleration profile from Eqn. 5, results in the plot of required torque versus time and plots of the states, θ , ϕ_1 and ϕ_2 , versus time as shown in Fig. 6. The torque curve in Fig. 6 represents the function that must be learned by the ANN.

Using the same assumptions specified for the simple system, initial estimates for the parameters required by the input transformation of Eqn. 6 were determined. Unfortunately ANN learning with a stretched desired input did not produce good results in this case. Neither adding units, varying the time scale parameters, nor varying the input scale parameters changed the resulting behavior. Additional input cues seem to be required.

ANN practitioners commonly use tapped delay-lines to provide an ANN with a state (and/or desired input) history. It is also common to tap the delay-line at every sample and apply all contiguous sample points to the ANN input. For the large delays that exist in the flexible link system, providing all contiguous sample points would result in a large number of input units and consequently a large number of weights. A variation on this theme is to tap the delay line at a limited number of points. Because of the nature of the desired position and the time-scale of the required force, three taps from a desired input delay-line were found to be sufficient. The offset times for the taps were chosen so that the inflection points of the desired motion inputs occurred at times of 0, 1, and 2 seconds. The resulting input set, found to be effective in the simulation, was,

$$\bar{y}_{in,ann} = [40y_d(t+1) - 20, 40y_d(t) - 20, 40y_d(t-1) - 20]. \quad (11)$$

A graphical representation of the input set is shown in Fig. 7.

F. Flexible Link ANN, Bang-Bang Accel.

A four layer ANN structure with three inputs and one output was used in an attempt to learn the force function in Fig. 6. The input set from Eqn. 11 was used to drive the ANN. The network parameters and learning algorithm were identical to the simple system case. The

differences from the simple system case were the number of units, the number of training cycles required (approximately 20,000) and the output scale factor. One example of the generated force and the corresponding state trajectories are shown in Fig.8. Just as in the simple system case, increasing the number of units improves the rendition of the force, which also reduces the residual vibration. Increasing the number of units beyond the 3-70-5-1 configuration did not, however, significantly reduce the error.

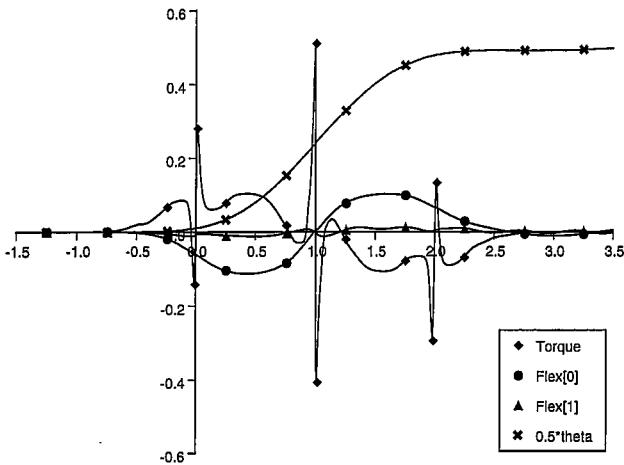


Figure 6, Force and states for flex link and bang-bang accel.

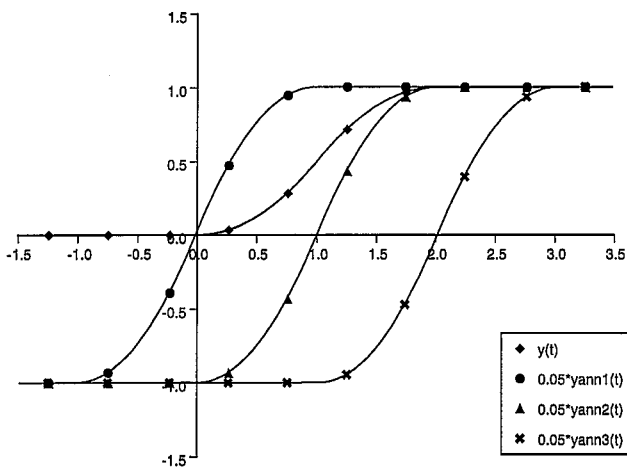


Figure 7, Example output of a sparsely tapped delay-line after the application of input scaling.

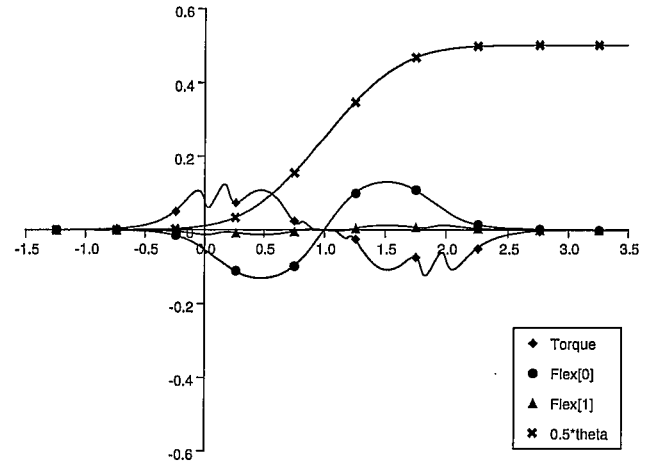


Figure 8, Results for sparsely tapped delay-line input, 3-70-5-1 ANN, bang-bang desired acceleration.

G. Flexible Link ANN Input/Output, Smooth Accel.

From a theoretical standpoint, the bang-bang trajectory from Eqn. 5 is convenient for at least two reasons. First, the bang-bang trajectory makes it is easy to explicitly solve the required convolution and second, the resulting force profile has drastic features. From a practical standpoint, however, a bang-bang trajectory introduces some difficulty. The first difficulty arises due to the truncated, assumed-modes model used to find the system transfer function. Because the model is truncated, there are natural frequencies that exist in the real system which are not represented by the model. To prevent excitation of the unmodeled modes, the desired acceleration profile should have no significant high frequency components (i.e., a smooth profile)[21]. The second difficulty arises due to the drastic features of the bang-bang induced force profile. The spikes can easily exceed the dynamic range of converters/amplifiers if the bang-bang trajectory is not carefully specified. Saturation of converters/amplifiers reduces performance and introduces significant high frequency components. A third difficulty also related to the drastic features of the bang-bang induced force profile is that the complexity of the ANN controller may be larger than necessary. The added ANN complexity causes problems for both learning (number of trials) and recall (processing requirements). To overcome these problems, the drastic features of the force profile are smoothed by replacing the sharp transition regions of the bang-bang desired acceleration with a third-order polynomial spline. The transition time for the polynomial is a free design parameter that may be tuned to eliminate high frequency vibration. Experiments for a single-link manipulator, similar to the model in Eqn. 9, found a transition time of approximately 0.2 seconds yielded good results[22]. The force profile and the corresponding state trajectories are shown in Fig. 9. As expected, the peak value of the force is lower than the comparable bang-bang induced profile. Also as expected, the force profile in Fig. 9 does not have discontinuities.

H. Flexible Link ANN, Smooth Accel.

Because the force profile in Fig. 9 does not have discontinuities, a three-layer network (e.g., one hidden layer) should be sufficient

to perform the mapping[18]. In this study, however, performances for three-layer networks in the single input case were disappointing. Three-layer networks ranging in size from 1-20-1 up to 1-200-1 produced poor results. Increasing the network structure to a four-layer network (e.g., 1-30-40-1) produced acceptable results, however, the required number of units was large (e.g., approximately 1300 weights). Because of this result, it is expected, although not investigated, that a one-input, three layer network with approximately 1300 hidden units would yield acceptable results. Results for the three-input, three-layer network were more encouraging. The three-input, three-layer network required an order of magnitude smaller number of weights than to the single-input case.

As in the previous two examples, a transformed representation of the desired tip motion was used as an input to the ANN. Both the scale-and-stretch and the sparsely tapped delay-line techniques were evaluated. Examples of the results obtained from the two techniques are shown in Figs. 10 and 11. Again, the network parameters and learning algorithm were identical to the simple system case. The differences from the simple system case were the number of units, the number of training cycles required (approximately 20,000) and the output scale factor.

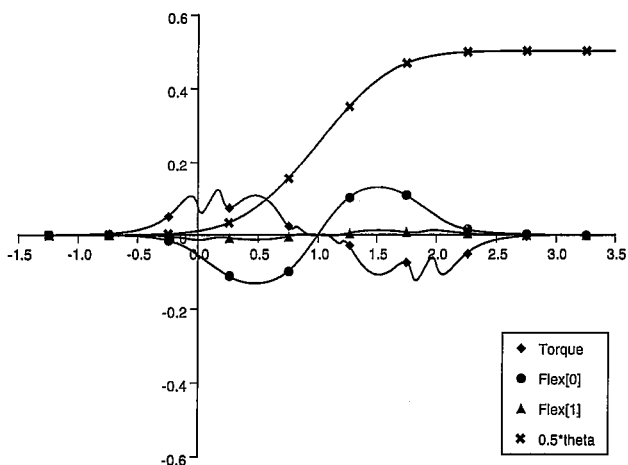


Figure 9, Force and States for Smooth Trajectory.

III. CONCLUSION

The results of the simulations indicate that a conventional ANN, of reasonable size is capable of learning the result of a convolution between one desired trajectory and a noncausal impulse response. The study concentrated on finding a transformation from the desired tip position to the ANN input that allowed the ANN to learn the force required by a single trajectory. The input set and the network configuration were found to impact both the learning speed and the resulting residual vibration. In this study, the sparsely tapped delay-line yielded a network with a small number of weights and appeared to be applicable to a wide range of force functions. The proposed methods for determining the trajectory delay (or equivalently, the desired input advance time) were shown to represent a good beginning point for finding the actual, required delay.

The control method in this study is strictly feedforward. The single trajectory learning condition is very restrictive and does not imply that the ANN has learned the plant inverse. Additionally, the learning method in this study is strictly off-line. An on-line learning method would greatly enhance the usefulness of the new method. The on-line learning methods suggested in [14] and [18] may be appropriate. More investigation is necessary before an inverse model ANN can be used to track a general trajectory or used inside a control loop. It is anticipated, however, that network structure, input set and delay requirements will be similar to those described in this investigation.

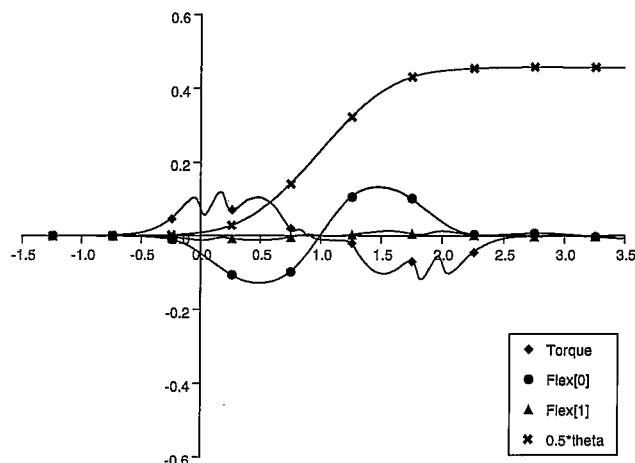


Figure 10, Scale-and-Stretch Input, 1-30-40-1 ANN.

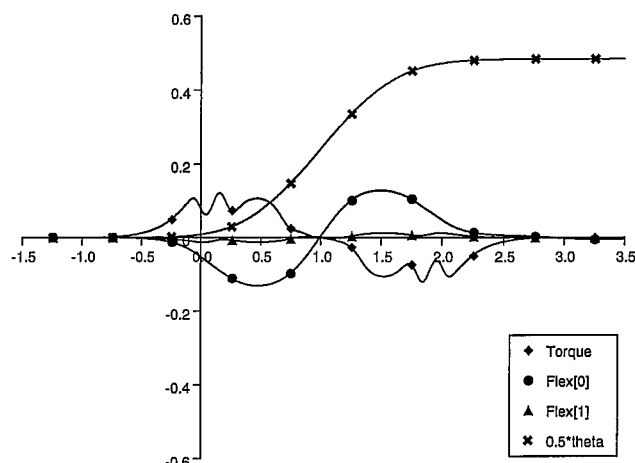


Figure 11, Sparsely tapped delay line input, 3-30-1 ANN.

IV. REFERENCES

- [1] Hunt, K., et. al., "Neural Networks for Control Systems - A Survey," *Automatica* 28(6), 1992, pp. 1083-1112.
- [2] Chang, S., Nair, S., "A New Neural Network Control Architecture for a Class of Nonlinear Dynamic Systems," *Proc. ACC*, 1993, pp. 79-83.
- [3] Newton, R., Xu, Y., "Neural Network Control of a Space Manipulator," *IEEE Control Sys. Mag.*, 13(5), Dec. 1993, pp. 14-22.
- [4] Askew, C., et. al., "A Neural Network Pattern Classification Approach for Payload Adaptive Regulation of Flexible Manipulators," *Proc. ACC*, 1993, pp. 2518-2519.
- [5] Arai, F., Rong, L., and Fukuda, T., "Trajectory Control of Flexible Plate Using Neural Network," *IEEE Intl. Conf. on Rob. and Auto.*, 1993, pp. 155-160.
- [6] Cetinkunt, S., Chiu, H., "Study of Learning Controllers for Tip Position Control of a Flexible Arm," *ASME Modeling and Cont. of Compliant Sys.*, DSC-31, 1991, pp. 15-19.
- [7] Takahashi, K. and Yamada, I., "Neural-Network-Based Learning Control of Flexible Mechanism," *Proc. ASME WAM, Intel. Cont. Sys.*, DSC-Vol. 48, 1993, pp. 95-104.
- [8] Register, A., Book, W., and Alford, C.O., "Artificial Neural Network Control of a Nonminimum Phase Mechanical System," submitted *IEEE Conf. on Rob. and Auto.*, (1996).
- [9] DeRusso, P., Roy, R., and Close, C., "State Variables for Engineers," *John Wiley & Sons, New York*, (1965).
- [10] Bayo, E., "A Finite Element Approach to Control the End-Point Motion of a Single-Link Flexible Robot," *Journal of Robotic Systems*, Vol. 4, No. 1, 1987, pp. 63-75.
- [11] Kwon, D.-S. and Book, W.J., *Proc. ASME WAM, Modeling and Control of Compliant and Rigid Motion Systems*, DSC-Vol. 31, Dec. 1, 1991, pp. 27-37.
- [12] Bayo, E., and Moulin, H., "An Efficient Computation of the Inverse Dynamics of Flexible Manipulators," *IEEE Intl. Conf. on Rob. and Auto.*, Vol. 2, 1989, pp. 710-715.
- [13] Widrow, B., "Adaptive Inverse Control," *Proc. of SPIE, Applications of Artif. Neural Nets.*, 1990, pp. 13-21.
- [14] Widrow, B. and Bilello, M., "Nonlinear Adaptive Signal Processing for Inverse Control," *Proc. World Congress on Neural Networks*, 1994, pp. (III-3)-(III-13).
- [18] Jordan, M. and Rumelhart, D., "Forward Models: Supervised Learning with a Distal Teacher," *Cognitive Science*, V. 16, N. 3, 1992, pp. 307-354.
- [16] Parthasarathy, S., Parlos, A. and Atiya, A., "Direct Adaptive Control of Process Systems Using Recurrent Neural Networks," *Proc. ACC*, 1992, pp. 63-65.
- [17] Eaton, J., et. al., "Stability of Neural Net Based Model Predictive Control," *Proc. ACC*, 1994, pp. 2481-2485.
- [18] Kurkova, V., "Kolmogorov's Theorem and Multilayer Neural Networks," *Neural Networks*, Vol 5., No. 3, (1992), pp 501-506.
- [19] Levin, E., Gewirtzman, R., and Inbar, G., "Neural Network Architecture for Adaptive System Modeling and Control," *Neural Networks*, Vol, 4, pp. 185-191, 1991.
- [20] Book, W.J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *Intl. J. of Robotics Research* 3(3), 1984, pp. 87-101.
- [21] Bayo, E., and Paden, B., "On Trajectory Generation for Flexible Robots," *Journal of Robotic Systems*, Vol. 4, No. 2, 1987, pp. 229-235.
- [22] Kwon, D.-S., "An Inverse Dynamic Tracking Control for Bracing a Flexible Manipulator," *Ph.D. Dissertation*, School of Mech. Eng., Georgia Inst. of Technology, 1991.