CONF-970430--8
ANL/RA/CP--91404

Paper to be presented at the 1997 Society of Computer Simulation Multiconference, Atlanta, Georgia, April 6-10, 1997

Neural Networks for Control of NO_x Emissions in Fossil Plants*

by

J. Reifman and E. E. Feldman
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

*Work supported by the U.S. Department of Energy, Energy Research Programs under Contract W-31-09-ENG-38

DISCLAIMER

DISCLAIMER

NEURAL NETWORKS FOR CONTROL OF NO_X EMISSIONS IN FOSSIL PLANTS

Jaques Reifman
Argonne National Laboratory
9700 South Cass Ave.
Argonne, Illinois 60439
E-mail: jreifman@anl.gov

Earl E. Feldman
Argonne National Laboratory
9700 South Cass Ave.
Argonne, Illinois 60439
E-mail: feldman@ra.anl.gov

KEY WORDS: Neural Networks, Fossil Plant, NO_x Emissions, Closed-Loop Control

ABSTRACT

We discuss the use of two classes of artificial neural networks, multilayer feedforward networks and fully-recurrent networks, in the development of a closed-loop controller for discrete-time dynamical systems. We apply the neural system to the control of oxides of nitrogen (NO_x) emissions for a simplified representation of a furnace of a coal-fired fossil plant. Plant data from one of Commonwealth Edison's (ComEd) fossil power plants were used to build a recurrent neural model of NO_x formation which is then used in the training of the feedforward neural controller. Preliminary simulation results demonstrate the feasibility of the approach and additional tests with increasingly realistic models should be pursued.

INTRODUCTION

Recent research (Narendra and Parthasarathy 1990) has demonstrated the potential of applying artificial neural networks (NNs) for the identification (modeling) and control of nonlinear dynamical systems. For process identification, NNs are valuable when the system is known in terms of its inputs and outputs and exact analytical models based on physical principles are unavailable or difficult to develop. For process control, NNs offer the advantage of being able to handle a large class of nonlinear control problems without requiring linear approximations of the system which often distorts the real problem. Because of these advantages NNs are currently being considered for modeling and control of NO_x emissions from fossil power plants.

Proprietary neural network-based systems, such as Gnocis (generic NO_x control intelligent system) being sponsored by the Electric Power Research Institute and NeuSIGHT being developed by Pegasus Technologies Corporation of Panesville, Ohio, are currently being tested on fossil power plants. Another system that was also tested in an actual plant and for which more in-depth technical information is available in the open literature is the one developed by Stone & Webster (Reinschmidt and Ling 1994). In Reinschmidt and Ling's approach, two multilayer feedforward NNs are used to provide open-loop control of NO_x emissions for steady-state plant conditions. Given a desired NO_x emission level, the NN system provides the settings of the furnace control variables that achieve the desired steady-state NO_x emission level after all transitory behavior has died out. In addition to providing only steady-state relationships with no feedback from the plant to the control system, their controller needs to be retrained off-line for every desired NO_x level and plant load.

In this work, we propose the combined use of two classes of NNs, multilayer feedforward networks and fully-connected recurrent networks, in the development of a closed-loop nonlinear controller for discrete-time dynamical systems. A feedforward network is used to represent the nonlinear controller and a recurrent network is used to represent the dynamical system in the training of the controller. We provide results of our initial investigation of applying the proposed NNs for modeling and control of NO_x emissions of a simplified representation of a coal-fired fossil plant. Unlike previous attempts, the proposed software performs closed-loop control with feedback from the plant to the controller and accounts for time-dependent information of the plant behavior.

CLOSED-LOOP CONTROLLER

The representation of the closed-loop controller used in this work is similar to the one proposed by Piche′ (1994) in which the dynamical system is composed of two basic components: a neural network controller and a plant model. Figure 1 illustrates the closed-loop representation with one-step delay for the simplified control problem used in this initial investigation where the controller and the plant model are serially arranged. The controller is represented by a feedforward multilayer neural network of the type proposed by Rumelhart et al. (1986). Given a desired target NO_x, demanded power, and current NO_x levels at time k, the controller adjusts the settings of the control variables, e.g., excess oxygen and burner tilts, such that the predicted NO_x at time k+1 tracks the desired target NO_x.

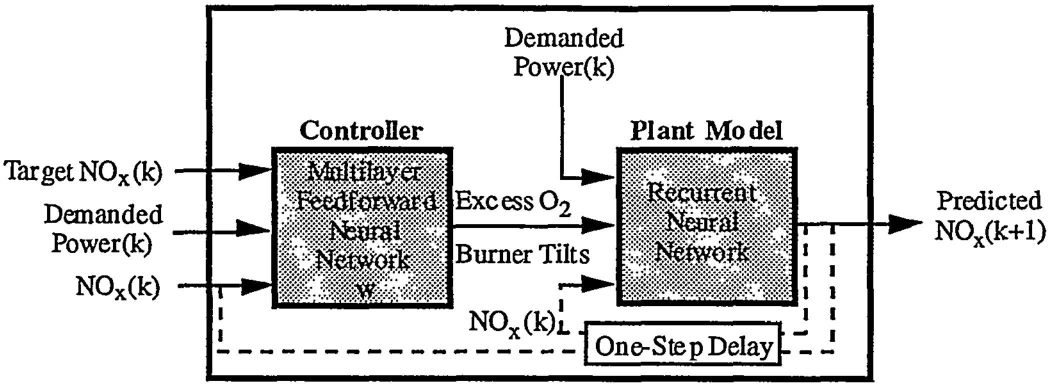

Figure 1. Closed-Loop Controller/Model Representation for Discrete-Time Nonlinear Dynamical Systems.

For representing the plant model when the dynamical system is unknown, Piche′ proposes the use of feedforward networks which are capable of approximating the plant dynamics through the use of feedback connections. The representation in Figure 1 lends itself to this proposition because the plant output $NO_x(k)$ is fed back into the plant model with one-time step delay in order to estimate $NO_x(k+1)$. However, comparison of feedforward network results with fully-recurrent networks, which implicitly represent time dependencies, indicated that feedforward networks may not be as accurate in predicting the dynamics of the plant behavior as fully-connected recurrent networks (Reifman et al. 1996). Therefore, here we propose the use of a fully-connected recurrent network to represent the plant model during the training phase and testing phase of the controller. When the system of Figure 1 is used for real-world control, the plant model is replaced by the actual plant.

NEURAL NETWORK PLANT MODEL

Unlike feedforward networks in which the present outputs depend solely on the present inputs, recurrent networks contain recurrent connections or feedback that together with time delays allow for the internal or implicit representation of time-dependent information. The fully-connected recurrent network proposed by Williams and Zipser (1989) was used in this study to represent the plant model in the closed-loop representation of the dynamical system in Figure 1.

The architecture of the recurrent network is one in which there are N external input units and M processing units. The processing units are fully connected with a unique weight between every pair of units (including itself), and also from each input unit to each processing unit. While in this architecture there is no distinction between hidden units and output units, not all processing units are "visible" to the external world. In general, only a subset J of the M processing units ($J \leq M$) are used as output units for which specified target values exist.

The learning algorithm is based on gradient-descent where at each iteration the network weights connecting each pair of units are updated such that the sum of the squares of the differences between the network predicted and target values is minimized. At each iteration the weights are updated proportional to their gradient components which are recursively calculated forward in time given that the gradient is zero at the first iteration. For a detailed description of the algorithm the reader should refer to Williams and Zipser (1989).

NEURAL NETWORK CONTROLLER

The controller is represented by a feedforward multilayer network of the type proposed by Rumelhart et al. (1986). However, because the dynamical system in Figure 1 contains feedback, the calculation of the partial derivatives of the dynamical system error E with respect to the feedforward network weights w are more difficult to calculate then when the feedforward network is a stand-alone module. The dynamical system error E is defined as the square of the difference between the target NO_x given as input to the system and the recurrent neural network predicted NO_x. By taking into account the temporal sequence of the calculations, the dynamical system in Figure 1 forms a set of ordered equations which need to be properly differentiated.

To provide the mathematical basis for computing the partial derivatives of a set of ordered equations, Werbos (1990) introduced the concept of *ordered* partial derivatives. For example, suppose we had a system governed by the following set of ordered equations:

$$x_2 = f(x_1)$$
$$x_3 = f(x_1,x_2)$$
$$x_4 = f(x_1,x_2,x_3)$$
$$\vdots$$
$$x_N = f(x_1,x_2,\ldots,x_{N-1}).$$

To calculate the "simple" partial derivative of x_4 with respect to x_2, we differentiate the equation for x_4 while holding x_1 and x_3 constant. However, when the *ordered* partial derivative of x_4 with respect to x_2 is calculated, x_1 is still held constant but x_3 is not. In the ordered partial derivative we also account for the fact that a change in x_2 causes a change in x_3 which, in turn, affects x_4. In essence, the concept of ordered partial derivatives provides a chain rule to recursively calculate the components of the gradient which are used to update the weights w of the feedforward network representing the controller. For a detailed description of the algorithm the reader should refer to Piche′ (1994) or Reifman and Feldman (1996).

PLANT DATA

Plant data from experiments conducted at the ComEd Will County Unit 3 (WCU-3) coal-fired electric power plant were used to develop the recurrent network model for NO_x emissions. WCU-3 is a 278 MWe Combustion Engineering tangentially-fired twin furnace where the coolant (water/steam) in this plant first passes through a superheat furnace, a high-pressure turbine, and then through a reheat furnace. During an 80-minute experiment, the plant was run in a load dispatcher mode where at every two minutes the demanded plant load and the furnace control variables were measured allowing for the collection of 41 consecutive sets of equally-spaced data points covering a limited range of plant operation.

The data collected at two-minute intervals include, NO_x emission, boiler master fuel flow rate (BM), which corresponds to the rate at which a mixture of coal and air is provided to the furnace, excess oxygen (O_2), the tilt of the burners in the superheat furnace (ST), and the tilt of the burners in the reheat furnace (RT). The values of NO_x, BM, and O_2 were measured for both furnaces combined. A lack of a more complete set of measured variables required that a few approximations be made. BM is used to represent demanded plant power and O_2, ST, and RT are the only variables used to control the formation of NO_x. These approximations are reasonable since fuel flow rate is proportional to plant power and out of a dozen or more furnace control variables, excess oxygen and burner tilts have been found in previous work (Adali et al. 1995; Reinschmidt and Ling 1994) to be strongly correlated with NO_x formation.

NEURAL NETWORK TRAINING

Since a model of the plant is required to train the neural network controller, we first need to select an architecture and train the recurrent network representing the plant. A 10-processing unit (M=10) network with five external inputs (N=5) was selected and trained to predict NO_x (Reifman et al. 1996). One of the 10 processing units served as an output unit (J=1) corresponding to the predicted NO_x at time k+1, and the three control variables, O_2, ST and RT, plus the demanded power, i.e., BM, and NO_x at time k were used as the five external inputs to the network. During the training phase, measured NO_x was used as one of the five inputs. After training, during operation, calculated NO_x at time k was fed back and used as input for predicting NO_x at time k+1.

By associating the plant input at time k with the plant output at time k+1, the 41 sets of data points collected formed a set of 40 input/output pairs. Out of the 40 input/output pairs, the first 20 were used for training the recurrent network and the last 20 for testing. Figure 2 shows the measured NO_x (solid line) and the network predicted values (circles) for the 40 points, which have a maximum difference of about 15%. The first 20 circles have plus signs inside them to indicate that these data points were used for training. This explains the excellent agreement between the measured and calculated values for the first 20 points. The value of all variables presented in this section and subsequent sections have been properly normalized and only their normalized values are shown. BM was normalized to the [0.2,0.8] interval and all other variables were normalized to the [0.0,1.0] interval.

The less than perfect match between measured and network predicted NO_x is attributed to the limited amount of data available for training and an incomplete set of control variables and state variables used to define the plant state. Therefore, the developed model should not be considered as a high-fidelity model of the actual plant. Instead, it should be considered as a model capable of representing the qualitative relationships between the represented plant inputs and NO_x formation. It should also be considered as a model capable of representing the dynamics of the physical plant since simulation results of dynamical tests indicated that the neural network model had captured the dynamics of the plant (Reifman et al. 1996). These capabilities of the plant model are sufficient (at this initial stage of the investigation) to determine the response of the neural network controller.

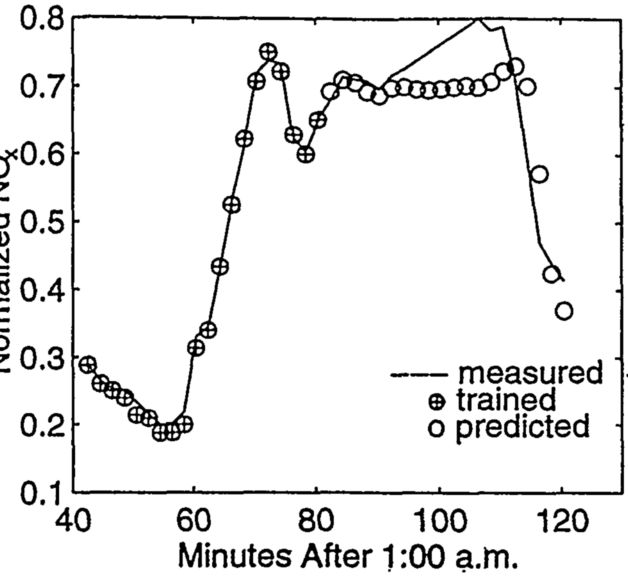

Figure 2. NO_x Measured and Predicted by the Recurrent Neural Network Model.

With the recurrent neural network model for NO_x emission developed, we then trained the multilayer feedforward network with the integrated dynamical system depicted in Figure 1 to drive the system from an arbitrarily selected initial state $NO_x(1)$ to a desired target NO_x for a given demanded power. In all training sessions, a three-layer network with three input units and three output units was used. The three inputs correspond to $NO_x(k)$, target NO_x, and demanded power and the three outputs correspond to values of the three control variables, $O_2(k)$, ST(k), and RT(k), at time step k. The number of units in the hidden layer was empirically determined and varied for different training sessions from five to twenty-five. Except for the input units, which were mapped by a linear function, all other units were mapped by a sigmoid function (Rumelhart et al. 1986).

In each training session, the feedforward network was trained to control the plant for specific regions of operation in the two-dimensional phase space defined by target NO_x and demanded power. To allow the neural network controller to map the entire specified region of operation, we first divided the specified region into 16 cells of equal size. Then, for each trajectory, i.e., for each sequence of steps that takes the system from an initial value $NO_x(1)$ to a final value $NO_x(k=k_{final})$, we sequentially selected values of target NO_x and demanded power in each one of the 16 cells, where the selection within a cell was randomly determined. During training, the selected values of target NO_x and demanded power were held constant for each trajectory. After each one of the 16 trajectories we calculated the error gradient and accumulated the weight update. After the 16 trajectories were calculated we checked if the convergence criterion $|NO_x(K) - \text{target } NO_x| \leq 0.01$ was satisfied for each trajectory. If satisfied, the training session was terminated. Otherwise, the weights were updated and a new sequence of 16 trajectories was calculated. The weights were updated using the conjugate gradient version of backpropagation discussed in Reifman and Vitela (1994). Since the conjugate gradient method dynamically optimizes the learning parameter and the momentum parameter, these did not enter as study parameters. In addition, for each training session, different sets of initial weights and number of units in the hidden layer were used. These degrees of freedom during the training stage allowed the weights to converge to acceptable non-local minimum values and the controller to find different control solutions or strategies for a given region of operation.

SIMULATION RESULTS

In order to illustrate the behavior of the closed-loop control system illustrated in Figure 1, we present the results of one of many simulated cases. For this case, the operational region was defined by a rectangular area with demanded power in the [0.6,0.8] interval and target NO_x in the [0.3,0.5] interval. For this operational region, numerous training sessions produced a few distinct control strategies. The strategy found most frequently (80%) was one in which only one, RT, out of the three control variables, O_2, ST and RT, was adjusted. From these results it seems that the training algorithm favors the simplest solution where only one variable is adjusted and that RT is preferred over the other two control variables. Figure 3 shows an example of this strategy found by the controller (lower graph) for arbitrary simulations of linear changes in demanded power and target NO_x (upper graph). The controller correctly adjusts the setting of reheat tilt RT while keeping the other two control variables fixed at 1.0 to allow the predicted NO_x to accurately track the target NO_x during the entire simulation.

Based on simulation tests using the plant model (Reifman and Feldman 1996) and the same operational region ([0.6,0.8] for demanded power and [0.3,0.5] for NO_x), we found that there are other feasible solutions with RT fixed and only one of the other two variables being adjusted. Namely, if O_2 is varied in the [0.0,1.0] interval and the two burner tilts are fixed at 1.0, the plant can reach any state of the operational region. However, the controller was never able to find such a control strategy.

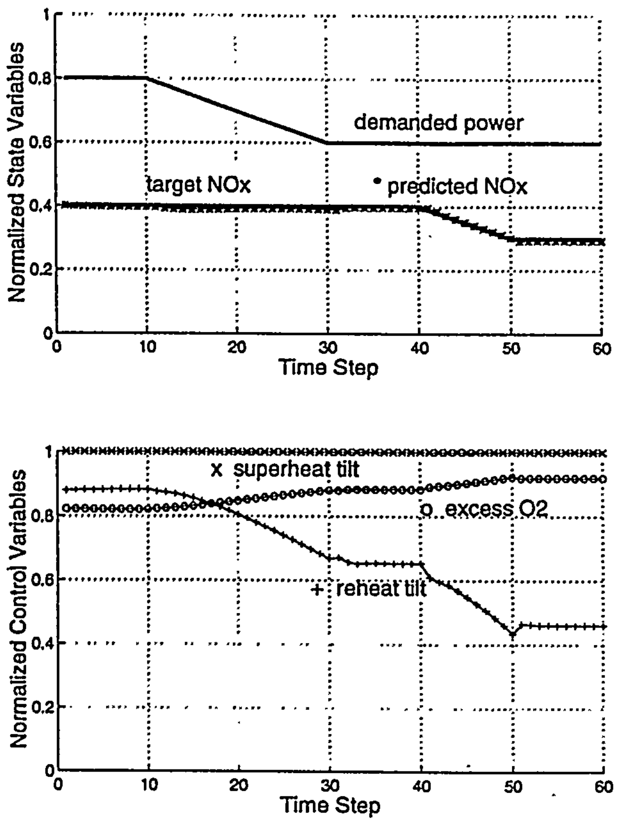

Figure 3. Typical Control Strategy Found By the Neural Network Controller for the Case Where Only Reheat Tilt Is Adjusted and Excess Oxygen and Superheat Tilt Are Fixed at 1.0.

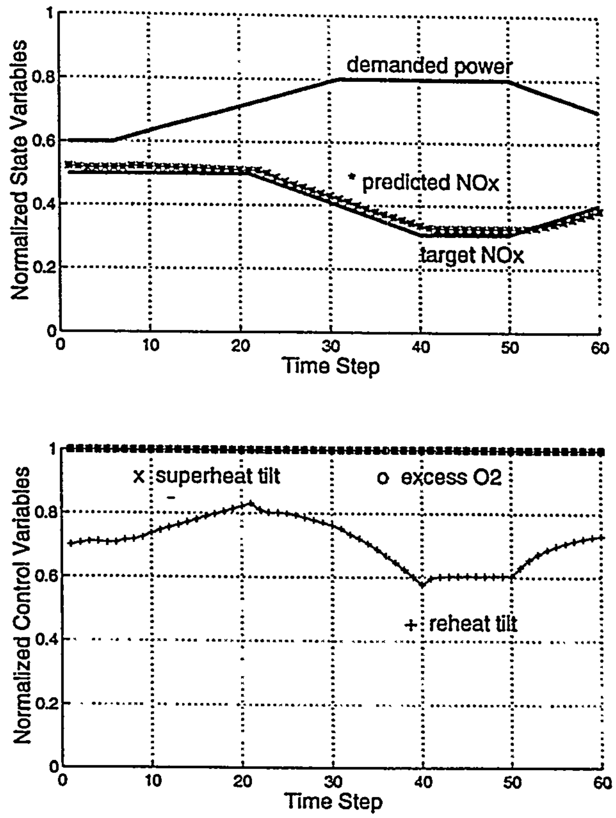

Figure 4. Typical Control Strategy Found By the Neural Network Controller for the Case Where Reheat Tilt and Excess Oxygen are Adjusted and Superheat Tilt is Fixed at 1.0.

In the remaining 20% of the training sessions where a solution was found (in many training sessions the weights converged to non-acceptable local minima), control strategies involving the combined use of RT and one of the two other control variables were found. In these cases RT was always the key controller with the other variable playing an ancillary role. Figure 4 illustrates such a case for an arbitrary simulation of linear changes in demanded power and target NO_x (upper graph) where the controller (lower graph) correctly adjusts RT and O_2 and keeps ST fixed at 1.0 to allow the predicted NO_x to accurately track the target NO_x.

No solutions were found in which the three control variables were simultaneously used. The arbitrary way in which neural networks find solutions precludes us from determining whether there are no such solutions for the operational region tried, or that such a solution is "harder" to obtain and could be found if we increased the number of training sessions. The fact that in 80% of the successful training sessions only the simplest solution was found may indicate that it is more difficult for the algorithm to find control strategies involving a larger number of variables. If this is the case, the usefulness of the approach as the dimensionality of the control variables is scaled up would be compromised. In addition, a large number of training sessions were not successful. Convergence to non-acceptable local minima was found to be a more significant problem when training dynamical systems than static systems.

SUMMARY AND CONCLUSIONS

This paper presents the results of our initial investigation in the application of two classes of artificial neural networks, multilayer feedforward networks and fully-connected recurrent networks, to control and model, respectively, discrete-time nonlinear dynamical systems.

The two networks are integrated into a closed-loop dynamical system that is applied to control NO_x emissions for a simplified representation of the furnace in a coal-fired fossil plant. With a limited set of plant data from one of ComEd's fossil power plants, we first train a recurrent network to serve as the model for the time-dependent formation of NO_x in the furnace. Then, we train a multilayer feedforward network to serve as the plant controller with the process modeled by the trained recurrent network. Once trained, the closed-loop dynamical system provides the settings of the furnace control variables such that the system predicted NO_x tracks the desired target levels of NO_x for each given plant demanded power.

Simulation results indicate that timely control maneuvers are provided by the neural controller such that for arbitrarily changing values of target NO_x and demanded power within specified regions of operation, the predicted NO_x tracks the target NO_x with a high degree of accuracy. For a given region of operation, distinct control strategies are found through different training sessions by changing the initial weights and number of units in the hidden layer of the neural controller. It was observed that the training algorithm tends to favor control strategies in which most of the control variables are fixed. Future work involving increasingly realistic models, with additional control variables represented and extensions made for optimization of the plant efficiency, should determine whether there are scale-up limitations inherent in the technique.

ACKNOWLEDGMENTS

This project was supported by the Laboratory Directed Research and Development program at Argonne National Laboratory. The authors would like to thank J. S. Herzau, D. G. Kubik, and K. E. Wanninger from ComEd for providing plant data and technical support.

REFERENCES

Adali, T., B. Baka, and R. Fakory. 1995. "NO_x and CO Prediction in Fossil Fuel Plants Using Artificial Neural Networks." In *Proceedings of the 1995 Society for Computing Simulation Multiconference* (Phoenix, Arizona, April 9-13), 1-10.

Narendra, K.S. and K. Parthasarathy. 1990. "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1: 4-26.

Piche', S.W. 1994. "Steepest Descent Algorithms for Neural Network Controllers and Filters." *IEEE Trans. Neural Networks*, vol. 5, no. 2: 198-212.

Reifman, J. and J.E. Vitela. 1994. "Accelerating Learning of Neural Networks with Conjugate Gradients for Nuclear Power Plant Applications." *Nuclear Technology*, vol. 106, no. 2 (May): 225-241.

Reifman, J., J.E. Vitela, E.E. Feldman, and T.Y.C. Wei. 1996. "Recurrent Neural Networks for NO_x Prediction in Fossil Plants." In *Proceedings of the 1996 Society for Computing Simulation Multiconference* (New Orleans, Louisiana, April 8-11), 89-94.

Reifman, J., and E.E. Feldman. 1996. "Identification and Control of NO_x Emission Using Neural Networks." Paper submitted to the Journal of the *Air & Waste Management Association*, December 1996.

Reinschmidt, K.F. and B. Ling. 1994. "Neural Networks for NO_x Control." In *Proceedings of the 1994 American Power Conference* (Chicago, Illinois, Apr. 25-27), vol. 56-I, 354-359.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986. "Learning Internal Representations by Error Propagation." *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. D. E. Rumelhart and J. C. McClelland, eds., The MIT Press, Cambridge, Massachusetts, vol. I, 319-362.

Werbos, P.J. 1990. "Backpropagation Through Time: What It Does and How to Do It." *Proc. IEEE*, vol. 78: 1550-1560.

Williams, R.J. and D. Zipser. 1989. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks." *Neural Computation*, vol. 1: 270-280.