AIAA-93-3273

Active Flow Control with Neural Networks

Xuetong Fan, Lorenz Hofmann & Thorwald Herbert
The Ohio State University
Columbus, Ohio

AIAA
Shear Flow Conference
July 6-9, 1993 / Orlando, FL

ACTIVE FLOW CONTROL WITH NEURAL NETWORKS

Xuetong Fan[*], Lorenz Hofmann[*] and Thorwald Herbert[†]
Department of Mechanical Engineering
The Ohio State University
Columbus, Ohio 43210, USA

Abstract

We conduct a conceptual study of active laminar flow control with neural networks to evaluate the feasibility of a "smart wall" which would combine micro-electro-mechanical sensors and actuators with an artificial neural network in a single layer of silicon. This first phase of our study shows that properly trained neural networks can establish the complex nonlinear relationships between multiple inputs and outputs which are characteristic of an active flow control system. Numerical simulations of flow-control systems with pretrained neural networks demonstrate almost complete cancellation of single and multiple artificial wave disturbances as they occur in transitional boundary layers. A natural disturbance signal with developing wave packets from a wind-tunnel experiment is also successfully attenuated. We conclude that active flow-control systems based on neural networks hold great promise for real-world implementation of transition control and ultimately can be extended to drag reduction in turbulent flow.

1 Introduction

Aircraft drag reduction by flow control has been intensively studied over the past decades. The motivation comes from the global awareness of natural resource conservation, the energy crisis, and the desire to reduce the cost of aircraft operation. A 10% reduction in total drag would represent savings of several million dollars in fuel cost for the airlines[1, 2]. Skin friction or viscous drag is considered a major obstacle in the development of advanced aerodynamic and hydrodynamic vehicles[3].

Reduction of the viscous skin friction owing to the development of the boundary layer over aircraft surfaces as the largest contribution to the total drag has received the most attention[2]. Our efforts are solely concerned with this aspect of flow control. The state of the technological development in this area has been reviewed by various authors, e.g. Bushnell[3, 4], Coustols & Savill[5]. The potential benefits and difficulties in achieving viscous drag reduction have been evaluated thoroughly and fully recognized in the literature and need not be repeated here.

The reduction of viscous skin friction drag can be achieved by two different approaches. The first approach takes advantage of the inherent low-friction characteristics of laminar boundary layers and attempts to maintain laminar flow over an enlarged part of the wetted area. The second approach directly deals with the control of the turbulent flow structure. Laminar flow control, defined as a process to delay transition to turbulence, is of fundamental importance for the understanding and development of flow-control methodology. In this first phase of our studies, laminar flow control is the primary subject. However, the ultimate goal will be the extension of the concepts for applications to turbulent flow control.

In general, laminar flow control can be approached in different ways. The first group of methods uses stability modifiers that alter the instability characteristics and delay transition by modifying the basic mean flow. As pointed out by Arnal[6], maintaining laminar flow is essentially a war against the inflection points in the mean velocity profile. The weaponry comprises techniques such as aerodynamic shaping, wall suction, and wall heating/cooling, etc. The theoretical basis and recent development of these techniques are reviewed by Reshotko[7], Gad-el-Hak[8], and more recently by Arnal[6]. Appropriate shaping of an aerodynamic body can produce a favorable streamwise pressure distribution and yield natural laminar flow (NLF). Wall suction can reduce the boundary layer thickness and modify the mean velocity profile to gain stability. Wall cooling/heating causes the viscosity to vary with distance from the wall to affect the mean velocity profile. These techniques depend on thorough understanding of instability and transition mechanism for proper control effects.

An alternative approach to laminar flow control is

[*] Graduate Research Associate

[†] Professor of Mechanical Engineering,
Senior Member AIAA

through dynamic interaction with the fluctuating flow. This approach is based on the existence of a region of relatively slow linear TS wave growth in two-dimensional boundary layers. The wave-superposition principle is valid in this region and can be exploited to achieve wave suppression or attenuation. Wave superposition has been actively pursued both numerically and experimentally. The procedure usually involves detecting the most unstable TS waves that develop either naturally or from an artificial source, and then trying to suppress their growth by introducing out-of-phase control waves. Techniques of introducing control waves include vibrating ribbons[9, 10] or wires[11], heating strips[12, 13, 14], wall motion[15], plate vibration[16], and periodic suction-blowing[17]. The work in this area has shown that disturbance waves existing in flow field can be attenuated successfully if counter-acting waves with proper phase and amplitude are superposed. However, to actively or adaptively determine the proper phase and amplitude of instant control action remains to be solved. Ladd & Hendricks[18] and Ladd[19] proposed an adaptive filter which achieved only limited success in active control of TS waves. Additional unsolved issues concern for example the control of 3D disturbances, wave packets, and secondary instabilities. These issues have seriously limited the utility of wave cancellation as a drag reduction scheme[20].

Laminar flow control through stability modifiers is purely passive and limited by the need for a stable system. Active control techniques promise greater potential in realistic applications since they can stabilize a passively unstable system. Active control involves the use of sensors, control devices, and actuators. The effectiveness of control is determined by two factors: how well the control device recognizes the need for action and how well it models the response of the flow field to the action.

Most contemporary attempts to actively control flows rest on solving the differential equations or model equations for the fluid motion. While such techniques may work in certain cases, the time scales and spatial scales in transitional or turbulent boundary layers would require excessive computational speed. To implement a wave cancellation scheme on the wing of a commercial airplane, it is necessary to detect location, amplitude, and spatial extent of wave packets. Assuming size and spacing of control elements are of the order of the wavelength in the wave packets ($\lambda = 3.5$ mm at $U_\infty = 280$ m/s), the implementation requires $8 \cdot 10^4$ control elements/m^2. To perform a single floating point operation on each element during the advective time scale $\tau = 2\lambda/U_\infty = 25\mu$s of the waves requires 3 Gflops/m^2 using a conventional serial computer. This estimate increases to 26 Tflops/m^2 for turbulent flow. Therefore, solving even a simple model of the Navier-Stokes equations on the fly appears illusionary.

As an alternative, more promising approach, we study the concept of a "smart wall" that combines two recently emerged technologies for active flow control through dynamic interaction with the fluctuating instability field. These technologies are artificial neural networks (ANN) and micro-electro-mechanical systems (MEMS). The sizes of micro-manufactured sensors and actuators are consistent with the scales of waves or turbulent wall eddies (170μ for our airplane). Neural networks – non-model-based highly nonlinear devices – are able to establish a nonlinear relation between multiple inputs and outputs by supervised or unsupervised learning. Neural networks built in analog hardware respond instantly in real time. Since MEMS and ANN are manufactured by a common technology, the integration of sensors, actuators, and control devices into a silicon coating appears well feasible.

In this paper, we discuss the initial phase of a feasibility study for such a "smart wall" which primarily addresses architecture, training, and performance of neural networks and the spacing of sensors and control units. This computational study prepares the ground for an experimental effort [¶] to prove the concept for the 2D boundary layer on a flat plate in a wind tunnel. We also envision application to the control of cross-flow vortices on a swept wing.

2 Concept of a "Smart Wall"

A smart wall consists of a regular two-dimensional arrangement of control units flush mounted in the wall or embedded in a surface coating. Each control unit consists of sensors, one actuator, and a control device. Sensors may or may not be shared by neighboring control units. The control device is an artificial neural network with multiple inputs and a single output. The proper relation between sensory input and output to the actuator is pretrained or established by self-adaptive learning. This generic description leaves a realm of choices to be made before the control of an actual flow can be attempted. The concept also imposes restrictions necessary for ultimate practical applications.

The sensors are constrained to gather information on the flow at the wall, either wall pressure or wall shear stress. While such sensors are commonly used in experiments and micro-mechanical versions are in development, the actuators pose a more challenging problem. Local heating, though easy to achieve, may be too slow and consume undesirable amounts of energy. Wall motion or local blowing/suction appear more at-

[¶] Conducted in cooperation with J. H. Haritonidis tractive although piezoelectric materials and existing micro-mechanical devices permit only very small displacements.

The control device is the least well-known component of the system. Generally, in comparable active-control systems an attempt is made to closely model the dynamical behavior of the system on the basis of nonlinear equations. With the choice of neural networks as control device, we head here in the opposite direction.

Artificial neural networks offer a variety of attractive properties that depend on the network architecture, i.e. the type, number, arrangement, and connections of the processing elements (PE) or neurons. Neural networks are highly nonlinear systems able to establish a nonlinear relation between multiple inputs and outputs by supervised or unsupervised learning. The learning process in essence adjusts the weights on the internal connections. Supervised learning requires sets of data that specify the desired output for a given input. This training method can be replaced by unsupervised learning algorithms. Such algorithms are able to extract meaningful features or variables especially from noisy input patterns. The network can be designed to capture those components that constitute the dominant part of the input variable. Unsupervised learning may also be key to adaptive control by training on the fly. Trained networks can be *a posteriori* analyzed, e.g. to eliminate processing elements with insignificant weights or to extract a mathematical model of the transfer function.

Various types of neural networks are able to match large amounts of input information simultaneously and to generate categorical output. These networks are widely used for pattern recognition, e.g. for the analysis of sonar signals or target classification. In our context, the pattern may be given by the wall pressure fluctuations created by a wave packet.

Neural networks are able to synthesize complex continuous functions of many input variables from sample input/output. Applications of this mode range from commercial noise reduction in EKGs and telephone communication to the forecasting of financial and economic data or adaptive position corrections for robots. Networks have a distributed and associative memory. Therefore, they give reasonable responses to incomplete or noisy input. Some architectures when properly trained even allow "generalization," i.e. a reasonable response to "unknown" input outside the learned data. The distributed memory also makes these networks fault tolerant. In contrast to digital computers, neural networks exhibit a graceful degradation as processing elements or connections fail. Moreover, special constrained training techniques can be developed to improve the fault tolerance.

The control by analog neural networks can be performed in real-time and completely removes the computational bottleneck of conventional control approaches. Even the computational demand of digitally simulated networks is very small since they operate in parallel. Thus, for networks cast in silicon, the processing requirements are independent of the number of control elements and reduce to modest 0.04 Mflops for laminar flow and 0.83 Mflops for turbulent flow.

Before the control system can be build in hardware, however, suitable network architectures must be designed, trained, and tested by simulation on digital computers. These simulations also serve to determine optimum number and spacing of the sensors and the need for connections between neighboring control units. The basic functionality of the control devices can be established assuming an ideal flow response equal to the controller output. Limited studies on the response of the real flow to the controller output can be performed using the parabolized stability equations. However, to fully capture the response of the flow to actuators, it is inevitable to perform scaled experiments in the wind tunnel. The scaling is primarily used to prolong the advective time scale. In a flat-plate experiment at $U_\infty = 6\text{m/s}$, a medium-range workstation [‖] permits up to 175 operations for each control unit in a 10×10 array, the approximate dimension of the experimental control systems. Provided a sufficiently simple network architecture can handle the control task, the workstation used for simulations can be connected directly to the experiment.

In the following, we report on the first phase of the study, the computer simulation of neural networks and control units with ideal flow response. Since successful work with neural networks is more a matter of experience than of rigorous strategies, and the literature offers many opinions and few proofs, various studies were conducted to explore general characteristics of neural networks. Some of these results are discussed before entering the area of wave cancellation.

3 Basic Studies on Neural Networks

Traditional expert systems incorporate knowledge through an explicit set of rules. The major advantage of neural networks is their ability to establish individual rules for each particular system. In supervised learning the network configures a particular connection weight mapping for each set of training data according to a specific training algorithm. As the network is repeatedly exposed to the same data set, the connection weights converge to optimal values, provided data set and train-

[‖] SGI Indigo R4000. 16 Mflops

ing algorithm are properly chosen. In recall mode, the trained network is then capable of returning correct output for each input stimulus.

The creation of a good neural network model of a given system require numerous choices concerning network architecture, training algorithm, and training data. Only the number of input and output channels is specified by the given system. The task of specifying the architecture, in particular the number of PEs and hidden layers and their connections, is simplified by use of the Professional II/Plus software from NeuralWare, Inc. This software also permits selection of a variety of training algorithms and output of the trainable or recallable network as C code.

The effectiveness of a neural network in any application depends on the proper selection of the training data. Typical sets of training data consist of 100 to 1000 combinations of numerical inputs to the network and the desired outputs. Inappropriately chosen training sets slow the training time, decrease the modeling accuracy, and result in a poorly trained network. The following two examples illustrate the wide range of system modeling capabilities of neural networks when proper stimuli are selected.

3.1 A Turbine Speed Controller

A turbine speed controller with proportional control is chosen to test the modeling proficiency of a neural network for a nonlinear feedback control system. As training data, 60 pairs each of rpm vs. time for four goal speeds are generated by an analytical model of the controller. The objective is to mimic the mechanical mechanism of the speed controller by a neural network.

The network architecture consists of two inputs (goal speed, time) and one output (response rpm) connected to a hidden layer with five PEs. The network is trained with the directed random search (DRS) training algorithm until the rms error is 0.005. This low error reflects that the turbine speed controller is well modeled by the trained network for the four goal speeds. Figure 1 shows the capability of this network to return a proper response for a goal speed outside the training set ("generalization"). The network achieves the desired rpm and accurately models the nonlinear relationships for the untrained data set. The success in modeling the turbine speed controller encourages attempting a more complex nonlinear model.

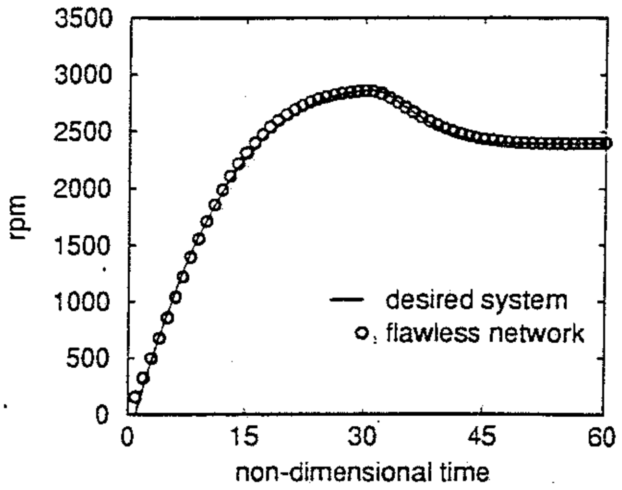

Figure 1: Network output vs. system output of a turbine speed controller

3.2 Lorenz Equations

The Lorenz equations are a three-dimensional system of ordinary differential equations that depend on three real, positive parameters σ, b, r:

$$\begin{aligned} \frac{\partial x}{\partial t} &= \sigma(y-x) \\ \frac{\partial y}{\partial t} &= rx - y - xz \\ \frac{\partial z}{\partial t} &= xy - bz \end{aligned}$$

The Lorenz equations describe a nonlinear dynamical system that exhibits chaos. For the constants $\sigma = 10, b = \frac{8}{3}$, and $r = 100$, the solution returns a periodic orbit known as the Lorenz attractor. Our goal is to model the highly nonlinear characteristics of the Lorenz attractor.

A fourth-order Runge-Kutta integration scheme is used to generate the numerical solutions to the Lorenz equations. Figure 2 shows the 2D projection of the attractor in the x, z plane. The network model entails inputs of x, y, z at time t and outputs of x, y, z at time $t + \delta t$, where δt is held constant. The network contains six PEs in the hidden layer. DRS is again employed as the training algorithm. Figure 3 illustrates the network output for the Lorenz attractor as seen in the x, z plane. The overall picture of the attractor is clearly different from the numerical solution in Figure 2, because it accumulates many small differences in the step by step prediction. However, the model achieves considerable accuracy for each single step, a task critical to flow control applications. Accurate modeling is also evident in the y, z plane. Similar results were reported by Welstead[21], who used the common back-propagation training algorithm and a larger number of PEs in the hidden layer. We conclude that relatively simple neural networks are capable of modeling highly nonlinear systems as long as proper training data are selected.

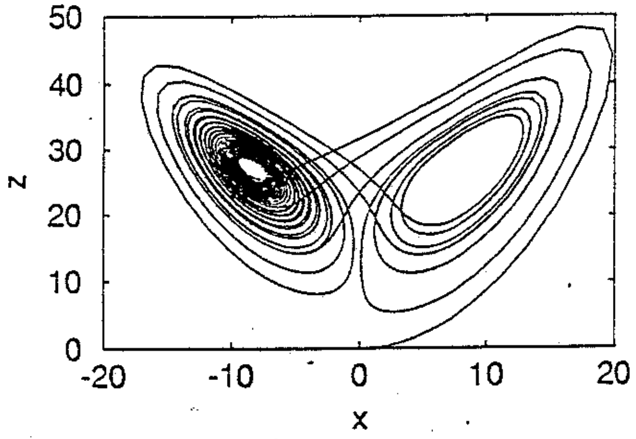

Figure 2: Lorenz attractor, system modeling in the x, z plane

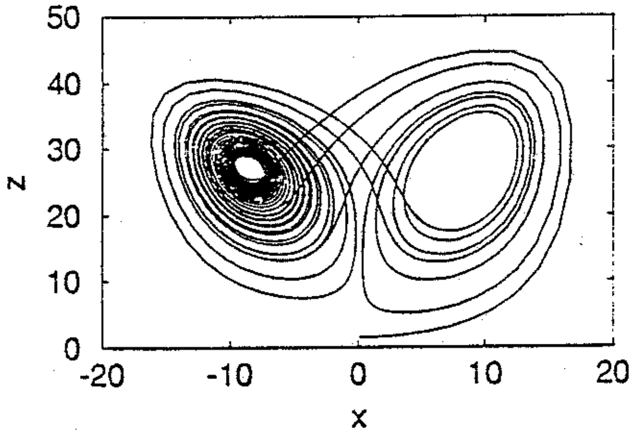

Figure 3: Lorenz attractor, neural network modeling in the x, z plane

Figure 4 shows the network architecture for the Lorenz attractor as displayed by the neural network software toward the end of training. Information propagates from the lower input layer of three PEs (x, y, z), through the hidden layer with six PEs, to the uppermost output layer of three PEs (x, y, z). Each box represents a PE. Each line connecting the various PEs represents the weighted connection between each PE. The size of the box is commensurate with its instantaneous output magnitude. A hollow box represents a negative value while a solid box represents a positive value. The three diagrams in the upper right show the confusion matrices for the three output PEs. Each matrix displays the correlation between actual and desired output. The overall rms error vs. iteration number is plotted in the lower right. The error is nearly constant at 0.0057. The use of rms error as a measure of modeling proficiency will be addressed shortly.

3.3 Training Strategies

Our extensive experience with the previous two systems suggests some comments on training data sets and training algorithms.

A training data set can contain any number of data points. Increasing the number of data points does not necessarily increase the network accuracy. However, concentrating data points at locations of greatest nonlinearity in the model assists the network in "learning" these more difficult patterns. Concentrating too many data points in linear regions of the model will bias the network toward these simpler patterns. Efforts to date in improving the training data set come at the expense of time for trial and error. Besides optimizing the training data set, the network accuracy can be improved by selection of a suitable training algorithm.

The most commonly used back-propagation training algorithm employs the method of steepest descent to minimize the network error. Back-propagation converges quickly, but is often trapped in local minima which is highly undesirable. On the other hand, a random search technique is assured of ascertaining the global minimum of a particular error function at major expense of convergence time. The directed random search technique is essentially an improved random search technique which "guides" the search and therefore reduces convergence time, yet still guarantees convergence to the global minimum[22]. Experience with the Lorenz attractor and the turbine speed controller shows the DRS training algorithm to perform extremely well in prediction type, nonlinear models.

3.4 Performance Measures

Analysis of the turbine speed controller and Lorenz attractor is mainly based on qualitative comparisons. The primary quantitative measure for determining the accuracy after training a particular neural network is the rms error. The rms error is not very sensitive to individual data points, but a low value is one indication of an accurately trained network. In addition to accuracy for a particular training set, our applications are concerned with the generalization capability for untrained data.

The generalization capability of a network increases as the number of PEs in the hidden layer decreases. A network with too many PEs in the hidden layer memorizes the training set, instead of learning particular trends. In our applications, it is not feasible to train the network for all possible inputs, and good generalization capabilities are highly desirable. Therefore, we have attempted by trial and error to train networks with the least number of PEs needed to achieve an acceptable error. Generalization in essence accounts for flaws in the input data.

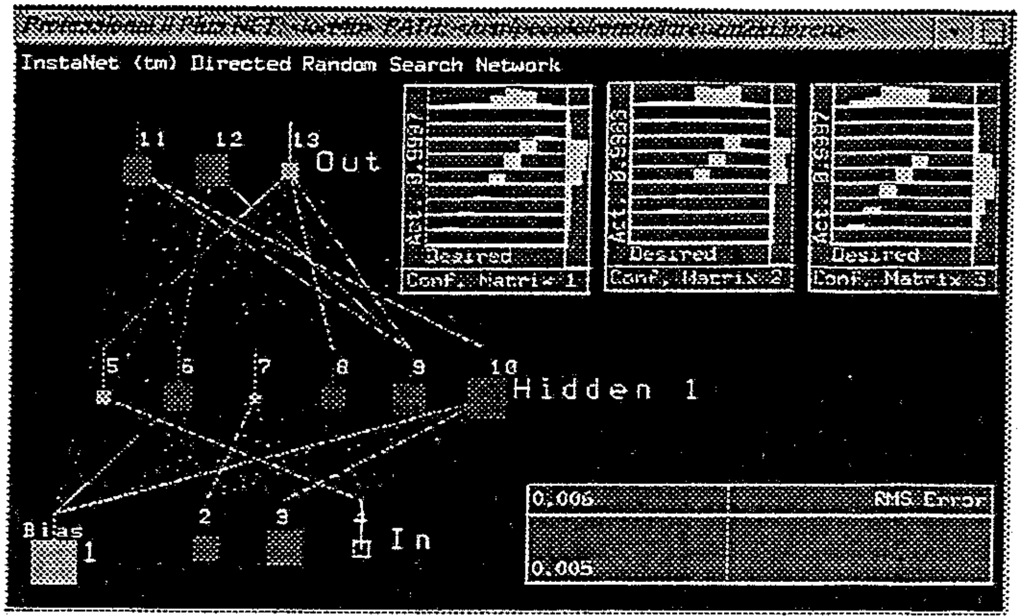

Figure 4: Neural network architecture via Neuralware Inc. software

3.5 Fault Tolerance

While a small number of PEs is desirable for generalization, simulation, and hardware implementation, it decreases the fault tolerance of the network to damage of PEs or connections. These flaws in the network architecture can be addressed by training the network to be more fault tolerant.

Fault tolerance accepts that an implemented system may not be perfect, and that measures are required to enable an operational system to cope with faults that remain or develop[23]. Fault tolerance via distributed architectures has been based on redundancy of hardware elements. A neural network can be trained fault tolerant instead, thus eliminating the need for extra PEs that would inevitably decrease the generalization capability. Fault tolerance of a neural network is based on maintaining performance in the network after removing a single hidden unit[24].

A neural network is typically trained for a specific number of hidden units. In training a network to be fault tolerant, one attempts to train that network for failures which might occur in those PEs in the hidden layer. Fault tolerance does not concern faults in the input or output layers.

The basic training algorithm for fault tolerance entails a constrained optimization problem. The goal is to minimize the error of the flawless network output, while not allowing the error caused by any faulted hidden element to increase by more than a prescribed amount. A directed random search technique with constraints was implemented and shown to converge to the optimal solution. Due to the additional constraints, a fault tolerant neural network will no longer converge to the low rms error of a standard network.

Figure 5 shows a standard network model of the simple function $y = x^3$. Note the undesirable output when the first or third hidden layer PE is removed. Figure 6 illustrates a fault tolerant neural network which pre-

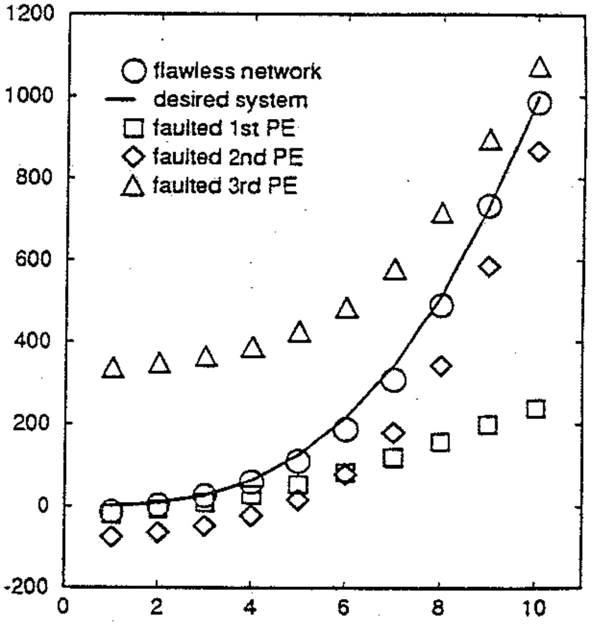

Figure 5: Standard network vs. system output of $y = x^3$

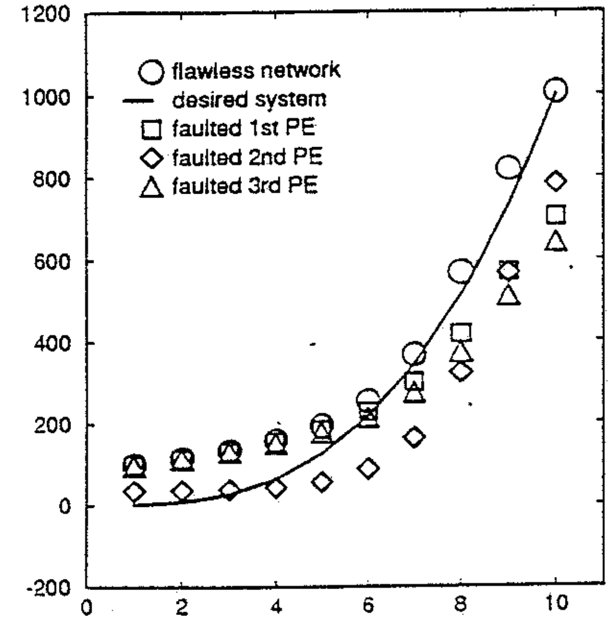

Figure 6: Fault tolerant network vs. system output of $y = x^3$

serves the integrity of the system regardless of faulted hidden layer elements. The additional network training constraint stipulated that the overall network rms error could not increase by more than 15% for any one flawed hidden layer PE. A flawless network models the system more accurately as the accepted percentage of degradation increases. Faulted networks, on the other hand, perform less desirably as the accepted percentage of fault tolerance is increased. Consequently, the required accuracy of a flawless network must be determined prior to training for fault tolerance. This new training algorithm for fault tolerance can be incorporated for predictive neural networks in applications where major performance degradation would be undesirable.

4 Control of TS Waves

Our study on the control of TS waves begins with artificially introduced waves. We assume that the variables to be controlled are the normalized flow quantities instead of the actuator movement as long as the flow receptivity to wall movement is uncertain. The transfer function between flow response and actuator movement will be our next research subject. After this transfer function is established, either by numerical calculation or experimental calibration, it can be easily incorporated by retraining the neural networks.

The schematic of the TS wave controller is shown in Figure 7, where "S_i" indicates sensors, "A" the actuator, and "SC" a scaling device. The dashed line representing the feedback signal from downstream sensor S_3 implies that feedback is disconnected. Currently, we consider only pretrained networks and on the fly training is not yet implemented. Two upstream sensors are necessary to provide the neural network with the sufficient information about the developing TS waves. In fact, a single-sensor system studied earlier proved to be a failure because the single sensor can not account for the propagating nature of TS waves. Use of a single sensor likely limited the success of adaptive filters in previous work[18, 19]. The schematic shown in Figure 7 actually represents one of the basic control units in the "smart wall" project. For the following case studies, this same control unit is utilized with different pretrained neural networks.

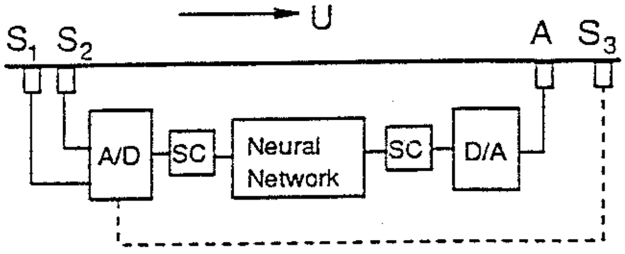

Figure 7: Schematic of the TS-wave control unit

4.1 Single Sinusoidal Wave

4.1.1 Single Wave with Constant Amplitude

The simplest case is to control a single sine wave of constant amplitude. A spatially propagating sine wave with frequency f, wavelength λ, and unit amplitude can be represented mathematically as $y = \sin(\omega t + \alpha x)$ where $\omega = 2\pi f$ and $\alpha = 2\pi/\lambda$. For complete cancellation of this wave at downstream position x_3, the neural network takes two inputs from upstream sensors at position x_1 and x_2, respectively, and is trained to generate an out-of-phase signal y_3' with respect to original signal at x_3, i.e. $y_3' = -\sin(\omega t + \alpha x_3)$.

A DRS network with 8 PEs in the hidden layer is built to model the nonlinear relationship between upstream sensor inputs and downstream flow response to the actuator. The two sensors are placed at $x_1 = 0$cm, $x_2 = 2/3$cm, respectively, and the actuator at $x_3 = 4/3$cm. The network is trained for $f = 80$Hz and $\lambda = 4$cm and converged to an rms error of 0.002. This well-trained network achieves almost complete cancellation when tested for the same frequency and wavenumber. Figure 8 shows the time series signal of the upstream sensor and the downstream residue after the control unit is activated. This network is also tested for generalization. The test results shown in Figure 9 indicate that the network performs satisfactorily for waves with trained wavenumbers but untrained frequencies, while it yields only partial suppression for waves with slightly different wavelengths.

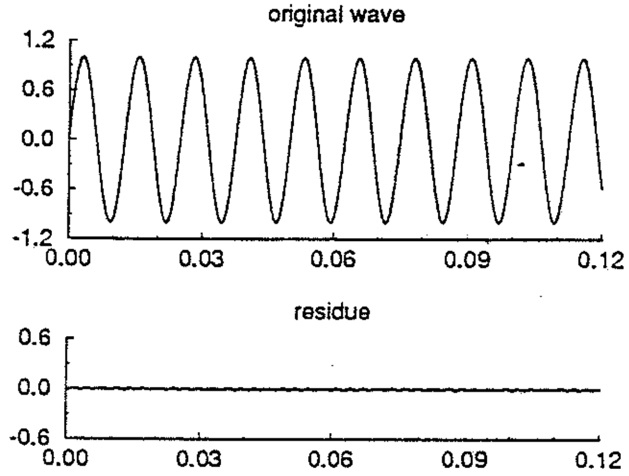

Figure 8: Neural Network control of a sine wave with $f = 80\text{Hz}$ and $\lambda = 4\text{cm}$

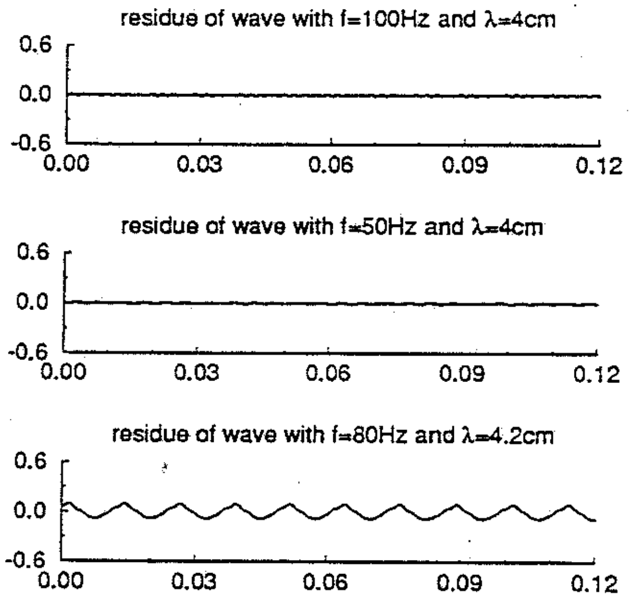

Figure 9: Test results for untrained frequencies and wavenumbers

Realizing the sensitivity of the neural network controller to wavelength, we train a more general network to account for a range of wavenumbers. We also reduce the spacing between the two sensors, and between the second sensor and the actuator, i.e. $x_1 = 0\text{cm}$, $x_2 = 1/3\text{cm}$, and $x_3 = 2/3\text{cm}$. A similar network is trained for $f = 80\text{Hz}$ and $\lambda = 3.5, 4.0, 4.5\text{cm}$ and converges to an rms error of 0.01. Figure 10 shows some results when this network is tested for some trained and untrained frequencies and wavenumbers. It is noted that this network offers almost complete cancellation for waves with trained parameters and good suppression for those outside the training range.

The above numerical simulation clearly shows the feasibility of control of sinusoidal waves with simple pretrained neural networks. It also demonstrates the generalization capability of neural network controller, as evidenced by the attenuation of waves outside the training regime.

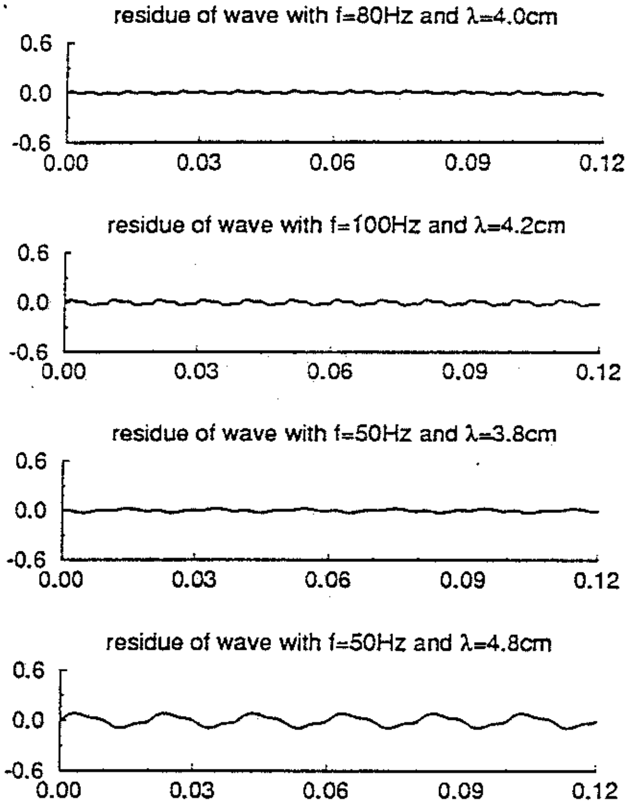

Figure 10: Test of a general network trained for a range of wavenumbers

4.1.2 Single Wave with Growing Amplitude

According to stability theory, TS waves with certain frequencies in the boundary layer will grow spatially and eventually lead to turbulence. Therefore, we attempt to control of a spatially unstable wave $y = e^{-\alpha_i x}\sin(\omega t + \alpha_r x)$. Figure 11 shows a spatially growing wave with $f = 80\text{Hz}$, $\lambda = 4\text{cm}$, and $\alpha_i = -0.1$. The exaggerated growth rate of $\alpha_i = -0.1$ emphasizes the unstable feature. A network with the same structure and sensor/actuator spacing is trained for $\alpha_i = -0.1$, $f = 80\text{Hz}$, and $\lambda = 3.5, 4.0, 4.5\text{cm}$ and converges to an rms error of

0.01. The test results shown in Figure 11 indicate excellent performance of the network. The network maintains satisfactory generalization capability.

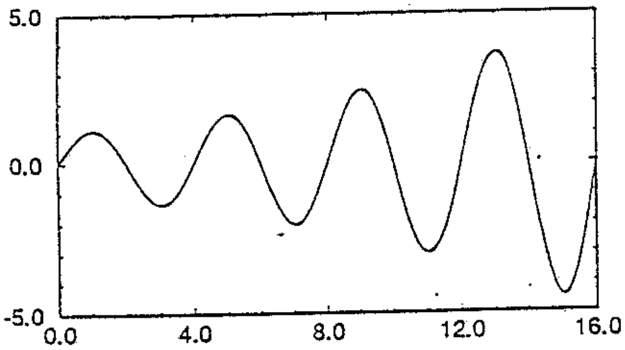

Figure 11: A spatially growing wave $y = e^{-\alpha_i x}\sin(\omega t + \alpha_r x)$

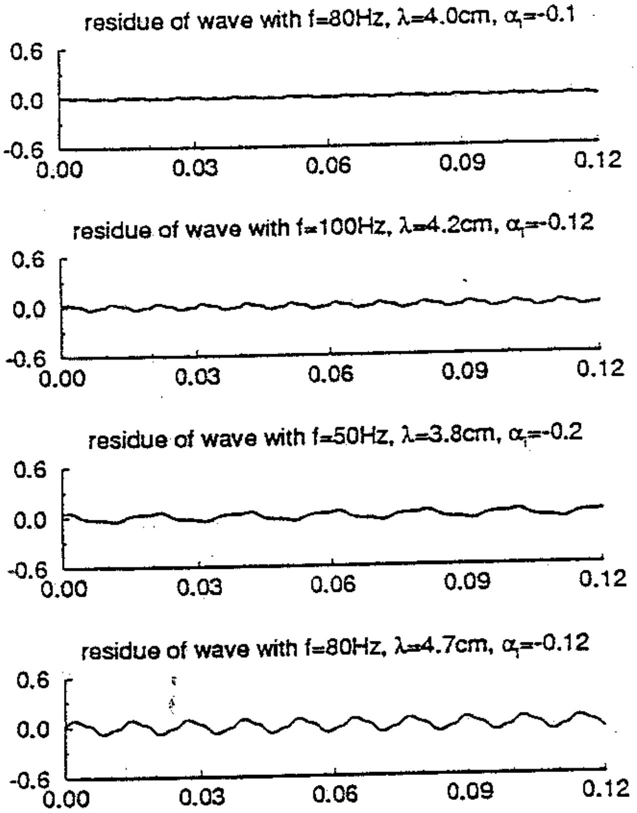

Figure 12: Control of spatially growing waves

4.2 Packet of Multiple Sinusoidal Waves

In real boundary layer flows, TS waves of various frequencies propagate at various speeds, growing or decaying. To approach this situation, we investigate the control of an artificial wave packet with multiple frequencies and wavenumbers. Because travelling waves can be represented as $y = F(x - ct)$, where $c = f\lambda$ is the wave speed, we expect a network which models waves with various speeds to be capable of controlling an artificial wave packet. Actually the general network trained in section 4.1.1 meets this requirement. We simply feed this network with signals from an artificial wave packet with four sinusoidal wave components with frequencies of 40, 50, 80, 100Hz and respective wavelengths of 3.5, 5.0, 4.5, 4.0cm. To our satisfaction, the network successfully attenuates this wave packet, as shown in Figure 13.

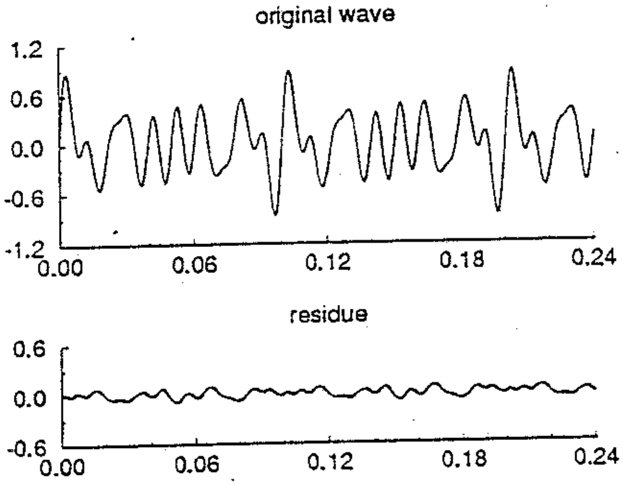

Figure 13: Control of an artificial wave packet

4.3 Experimental Wave Packet

Finally, we attempt to control actual wave packets as they appear in time series recorded in a wind tunnel. Detailed experimental data simultaneously obtained with hot wires, wall pressure probes (microphones), and hot films in a flat-plate boundary layer were kindly provided by Dr. Jim Kendall (JPL). In this wind tunnel experiment, the free stream velocity is 11.6 m/s and the turbulence level is 0.2%.

The signal recorded by microphone sensor C10 at a distance of 0.50 m from the leading edge contains numerous wave packets, as shown in Figure 14.

A neural network is trained to cancel a travelling sine wave with approximately the same speed as the wave packets in the time series. It is then applied to this experimental signal. Figure 15 shows very good attenuation of the wave packets in the signals. It is interesting to note that the noise level is reduced as well.

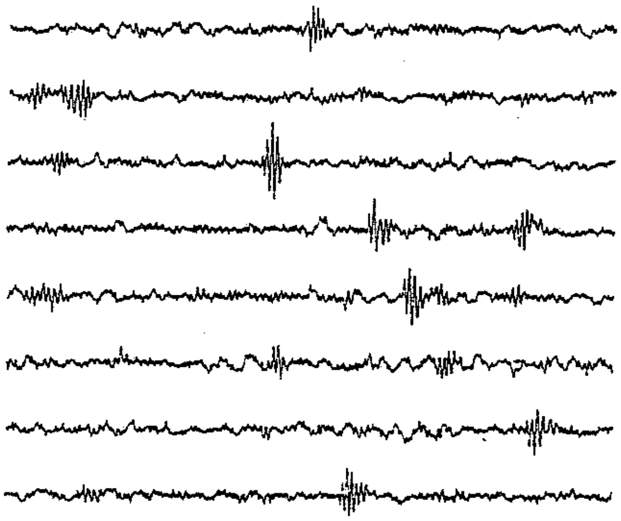

Figure 14: Microphone signal from experiment

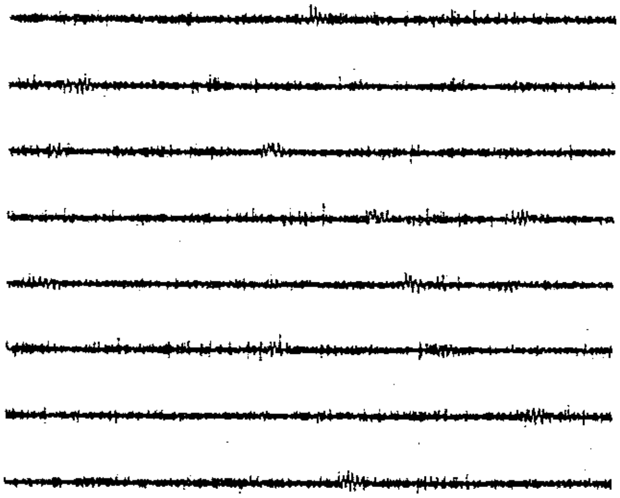

Figure 15: Residue of the experimental signal

5 Conclusions and Outlook

We have reported on the first phase of a conceptual study of a "smart wall" which combines artificial neural networks and micro-electro-mechanical systems for active flow control. Basic studies on neural networks verified their capability of modeling nonlinear relationships between multiple inputs and outputs. We also developed a new training algorithm for network fault tolerance.

As the primary goal of the current effort, we developed an active laminar flow control method based on neural networks and the wave superposition principle. Computer simulations of control of artificially induced simple waves, growing waves, and wave packets demonstrate substantial success in canceling or attenuating wave disturbances with a single control unit. The generalization capability of neural networks was also verified in these exercises. Finally, we applied the same control method to an experimental time series containing natural wave packets as they appear in a flat-plate boundary layer. We achieved excellent suppression of the wave packets, as well as reduction of the turbulent noise level.

The proposed active flow control method with neural networks appears to hold great promise in real world applications. An an important issue, we will next investigate on the fly training of neural network control systems with chained control units. The task of suppressing disturbances is then shared by a group of identical control units. An individual unit, trained in situ, only needs to partially suppress the total disturbance, a situation quite different from the study above. As soon as the instrumentation is completed, experimental evidence will be gathered to support our simulations. We will also study the flow response to actuators and include the transfer function into the training data.

Acknowledgment

We acknowledge fruitful discussions with J. Haritonidis, L. Mack, and J. Kendall. This work is supported by the Air Force Office of Scientific Research under Contracts AFOSR-91-02-62 and F49620-93-1-0135 (TH & XF) and by an AASERT Fellowship F49620-92-J-0339 (LH).

References

[1] Robert, J. P. 1992 "Drag Reduction : An Industrial Challenge," AGARD Report 786, Paper No. 2.

[2] Thomas, A. S. W. 1984 "Aircraft Drag Reduction Technology - A Summary," AGARD Report 723, Paper No. 1.

[3] Bushnell, D. M. 1990 "Introduction," in: *Viscous Drag Reduction in Boundary Layers*, Progress in Astronautics and Aeronautics, Vol. 123, pp. xi-xv.

[4] Bushnell, D. M. 1992 "Aircraft Drag Reduction," AGARD Report 786, Paper No. 3.

[5] Coustols, E. & Savill, A. M. 1992 "Turbulent Skin-Friction Drag Reduction by Active and Passive Means," AGARD Report 786, Paper No. 8.

[6] Arnal, D. 1992 "Boundary Layer Transition: Prediction, Application to Drag Reduction," AGARD Report 786, Paper No. 5.

[7] Reshotko, E. 1984 "Laminar Flow Control: Viscous Simulation," AGARD Report 709, Paper No. 8.

[8] Gad-el-Hak, M. 1990 "Control of low-speed airfoil aerodynamics," *AIAA Journal*, Vol. 28. No. 9, pp. 1537-1552.

[9] Thomas, A. S. W. 1983 "The control of boundary-layer transition using a wave-superposition principle", *J. Fluid Mech.*, Vol. 137, pp. 233-250.

[10] Pupator, P. T. & Saric, W. S. 1989 "Control.of Random Disturbance in a Boundary Layer," AIAA-89-1007.

[11] Milling, R. W. 1981 "Tollmien-Schlichting wave cancellation," *Phys. Fluids*, Vol. 24, No. 5, pp. 979-981.

[12] Liepmann, H. W., Brown, G. L. & Nosenchuck, D. M. 1982 "Control of laminar-instability waves using a new technique," *J. Fluid Mech.* Vol. 118, pp. 187-200.

[13] Liepmann, H. W. & Nosenchuck, D. M. 1982 "Active control of laminar-turbulent transition," *J. Fluid Mech.* Vol. 118, pp. 201-204.

[14] Kral, L. D. & Fasel, H. F. 1991 "Numerical Investigation of 3-D Active Control of Boundary Layer Transition," *AIAA Journal*, Vol. 29, pp. 1407-1417.

[15] Breuer, K. S., Haritonidis, J. H. & Landahl, M. T. 1989 "The control of transient disturbances in a flat plate boundary layer through active wall motion," *Phys. Fluids A*, Vol. 1, No. 3, pp. 574-582.

[16] Gedney, C. J. 1983 "The cancellation of a sound-excited Tollmien-Schlichting wave with plate vibration," *Phys. Fluids A*, Vol. 26, No. 5, pp. 1158-1160.

[17] Biringen, S. 1984 "Active control of transition by periodic suction-blowing" *Phys. Fluids A*, Vol. 27, No. 6, pp. 1345-1347.

[18] Ladd, D. M. & Hendricks, E. W. 1988 "Active Control of 2-D Instability waves on an axisymmetric Body," *Experiments in Fluids* Vol. 6, pp. 69-70.

[19] Ladd, D. M. 1990 "Control of natural laminar instability waves on an axisymmetric body," *AIAA Journal*, Vol. 28, No. 2, pp. 367-368.

[20] Thomas, S. W. A. 1990 "Active wave control of boundary-layer transition," in: *Viscous Drag Reduction in Boundary Layers*, Progress in Astronautics and Aeronautics, Vol. 123, pp. 179-202.

[21] Welstead, S. T. 1992 "Neural Network Modeling of Chaotic Dynamics in Nuclear Reactor Flows," *Transactions of the American Nuclear Society*, Vol. 65, pp. 210-211.

[22] Baba, N. 1989 "A New Approach for Finding the Global Minimum of Error Function of Neural Networks," *Neural Networks*, Vol. 2, pp. 367-373.

[23] Lee, P. A., Anderson, T. 1990 *Fault Tolerance Principles and Practice*, Springer-Verlag.

[24] Neti, C., Schneider, M., Young, E. 1992 "Maximally Fault Tolerant Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 14-23.