FINAL ID NO: F4.4-M.1.06

NEURAL NETWORKS FOR CONTROL, IDENTIFICATION, AND DIAGNOSIS

H. E. Rauch and D. B. Schaechter

Lockheed Palo Alto Research Laboratory, Palo Alto Ca 94304

ABSTRACT

Advances in the theory and technology of artificial neural networks provide the potential for new approaches to the problems of control, identification, and diagnosis for large, complex systems. However, these approaches must be validated for specific applications before they can be exploited effectively. Because of the unique capabilities they offer, neural networks should play an important role in space exploration systems operations. After a brief introduction to neural networks is presented, some applications of neural networks to identification and control of space systems are described and discussed. They span the spectrum of relatively straightforward to rather complex applications. An explanation of how neural networks can be applied to such important tasks as fault diagnosis and accommodation is presented. Neural networks are shown to be part of the hierarchy of intelligent control where a higher order decision element monitors and supervises lower order elements for sensing and actuation.

INTRODUCTION

The Report of the United States National Commission on Space has projected a series of manned missions to be pursued well into the Twenty-First Century. In order to complete these missions, it is necessary to develop a life support system capable of assuring human extraterrestrial existence over long mission duration. In anticipation of this need NASA is supporting the development of a Controlled Ecological Life Support System (CELSS). The goal of the CELSS program is to develop a regenerative life support system that relies on the abilities of organisms which use photosynthesis to produce food and oxygen from carbon dioxide and water. One phase of the CELSS program constitutes planned space flight experiments, involving free-flyers, and technology test facilities associated with the Space Station .

The planned CELSS flight experiments will require control systems that are capable of meeting stringent requirements over long periods of time, while operating autonomously and under a great deal of uncertainty. After the establishment of a system operating regime, the management of the overall control system will perform tasks such as monitoring system performance, monitoring the health condition of the system, coordination of operational subsystems, diagnosis of faults, and coordination of maintenance and repair. As much as possible, the operation should be autonomous, with the capability of human override if necessary.

Meanwhile, controllers for the operational subsystems continue to assure performance, thereby maintaining stability and tracking specifications. For example, large-scale space antennas must be able to operate successfully under a variety of unforeseen conditions. Large segmented mirrors must maintain a high geometric precision in the presence of thermal, gravitational, and dynamic loads. Light-weight, segmented, robot manipulators must move quickly and accurately while experiencing variable loading demands and trajectory requirements. The operational subsystems will also require control systems that are capable of meeting stringent and flexible performance requirements for identification, control, and diagnosis.

Neural networks have received widespread attention recently, particularly in the fields of pattern recognition, signal processing, and optimization. Only recently has attention been focused on applications of neural networks to real-time identification and control. Work is progressing on applying neural networks to fault detection, isolation, and control reconfiguration (FDIR). Neural network approaches for autonomous operation in space flight experiments can be evaluated in a step-by-step program designed to explore the most straightforward applications first, followed by more complex and demanding applications. In this way, knowledge and experience can be gained in an orderly manner while enabling early demonstrations of the new technology in simpler applications. Neural network applications for complex space systems might be envisioned in the following categories:

1) Identification -- Indicate specific parameters associated with observed system operation, and adjust learned representation of system characteristics in response to new measurements.
2) Adaptive Control -- Issue actuator commands in response to observed or estimated system states, and adapt responses to improve performance of the controlled system.
3) Monitoring and Classification -- Determine normal or abnormal system performance based on learned characteristics.

4) Reconfigurable Control -- Reconfigure input-output topology to recover acceptable system performance in the presence of failures of actuators or structural members.

NEURAL NETWORK BACKGROUND

The artificial neural network can be treated as a special purpose computer which is configured to represent a nonlinear mapping implied by empirical data. The mapping is implemented using a large number of individual processing elements (sometimes called neurons)with a dense network of interconnections. By defining proper processing functions for each element and proper weights for each interconnection, it is possible to solve difficult problems quite rapidly. Examples of such problems include function optimization and realization of nonlinear point-to-point mapping. Most importantly, neural networks have the capability to learn, so that performance can improve with time and adapt to changing stimuli.

In order to solve a practical optimization problem using a neural network, it is necessary (1) to choose an architecture, including the interconnections between individual processing elements, and (2) to choose the associated weights for the interconnections. The architecture is often chosen from experience or from some standard architecture. Under certain conditions the weights can be chosen based on the problem to be solved. However, in most cases it is necessary for the neural network to learn appropriate weights in a training procedure based on many iterations.

Narendra and Parthasarathy [1] give an excellent explanation of how neural networks can be used in identification and control of dynamic systems with nonlinearities. Their examples involve discrete time models of single-input single-output systems with memory, i.e., the model includes historical values of input and output. For identification, the input and output values of the system are fed into a multi-layer neural network. For control, measured output values and the desired output values are fed into a neural network. Back propagation is used to train the network to improve performance.

The most popular neural network architecture is the multi-layer feed-forward topology where the weights are adjusted by back-propagation [2]. Back-propagation can be used when there is some error function which is to be minimized, such as the error in correctly identifying characteristics of a system. Back-propagation is used during each step of training to calculate the gradient of the error with respect to the weights, and to adjust the weights slightly to reduce the error. Each gradient step is small, so versions of the same input-output data from the system might be presented to the network many times. After a large number of steps (perhaps thousands) the back propagation algorithm should approach a satisfactory set of weights. As new information is presented to the neural network, the weights can be updated so there is the possibility for improvement through continuous learning.

However, a disadvantage of back propagation is that it can take a large number of iterations to converge to the desired solution. An alternative to back-propagation that has been used in classification is the Probabilistic Neural Network (PNN) developed by Specht [3,4] which involves instant, one-pass learning and can be implemented directly in neural network architecture. A similar one-pass neural network learning algorithm also developed by Specht is called the General Regression Neural Network (GRNN) [5]. The GRNN shows promise for identification and control functions because it eliminates the extensive training period characteristic of most back propagation schemes by providing instant, one-pass learning in a growing network. In fact, the GRNN learns by allocating new network elements, rather than by adjusting a fixed number of internal parameters over several iterations, as in back-propagation networks.

The most difficult example investigated by Narendra and Parthasarathy [1] (with five inputs and one output) required over 100,000 iterations using back propagation to represent the nonlinear system function adequately. The same example required only 1000 input-output pairs using the General Regression Neural Network [5] to represent the nonlinear system function to the same degree of accuracy, and required only 100 input-output pairs to give an adequate, but slightly less accurate, representation. Given the potential for rapid training and retraining, the GRNN not only offers potential improvements in control synthesis and design time, it also increases the feasibility for using neural networks for real-time control and reconfigurable control.

A major benefit of neural networks derives from the tremendous computational speed that can be achieved by the massively parallel architecture of simple processors. An example of analog neural hardware used to control a deformable mirror in real-time is the Programmable Analog Neural Processor (PANP) developed by Lockheed [6]. The processor hardware contains 256 neurons (multipliers) and 2048 programmable synoptic weights (5 bits) operating in parallel. The bandwidth of the analog hardware is 90K Hz, and it can process analog inputs and outputs at that rate. The control loop for the deformable mirror has 42 sensors and 21 actuators. The closed loop system with the analog hardware controller can operate at a bandwidth (sample rate) of 173 Hz. The settling time of the closed loop system is 5 milliseconds, and the accuracy of the mirror control is 0.067 waves. Breadboard hardware has been tested successfully for a second generation analog processor based on commercially available chips which is more accurate (8 bits) and has a wider voltage range.

For an introduction to computing with neural networks, see the September-October 1990 Special Issues of the IEEE Proceedings [7,8]. For further explanation of control using neural networks, see the April 1990 Special Issue of the IEEE Control Systems Magazine [9] which contains 11 papers showing how to apply neural networks to identification and control. The April 1992 Special Issue on Neural Networks in Control Systems [10] contains 7 more recent papers on identification and control including applications to fault detection . The book Artificial Neural Networks [11], presents paradigms, applications, and hardware implementations.

EXAMPLE OF A NEURAL CONTROLLER

The example described here is based on a recent paper [12] which illustrates some control applications using the General Regression Neural Network. GRNN is a memory-based neural network with a straightforward implementation and a one-pass rapid learning algorithm that uses a highly parallel architecture. The algorithm provides smooth transitions between observed values even with sparse data in a multidimensional measurement space. Three significant advantages of GRNN are: 1) the network learns in one pass through the data and can generalize from examples as soon as they are received, 2) the network converges to the conditional mean as additional samples are received, and 3) the network has a clustering version which limits the number of nodes and provides an optional mechanism for ignoring old data.

For initial analysis, it is sometimes more convenient to implement neural network algorithms in software rather than hardware. Consequently, the GRNN has been implemented in software (in the C language) and integrated with the MATRIXx commercial software package for computer aided control system design. The software implementation of the GRNN algorithm is completely general so that the user can select the number of inputs, the number of outputs, and the number of training samples. The software has three operational modes: (a) learning, where training data is incorporated into the neural network; (b) standard operation, where the neural network recalls the training data for modeling, estimation, or control; and (c) adaptive operation, where new measurements are incorporated to update the training data while the training data is recalled for operation.

The neural network can perform modeling, estimation, or control. In particular, the neural network can emulate the dynamic system (modeling), can determine the state of the dynamic system (estimation), or can control the dynamic system (based on selected control strategies). The state of a nonlinear dynamic system can be considered the set of internal variables which completely describe the future output of the system when given the subsequent control inputs. Modern linear control involves these same three functions: (1) modeling, sometimes called identification, which develops a linear model based on control inputs and system outputs; (2) estimation, which determines the state of the system; and (3) control, which develops a linear control law to minimize loss.

The neural network performs a nonlinear mapping from measured input values to output values. As such, it is ideal for identification of the input-output transfer function of an arbitrary system. Training data consists of a set of typical inputs and corresponding outputs. In operation, the neural network receives input, and produces outputs that are consistent with previously learned data. Training may continue during operation, providing a system identification which adapts continuously.

Successful implementation of the learning mode requires samples which span the space of control inputs and outputs. If the number of training samples is limited, a carefully planned training phase is essential. When the training samples are based on random inputs and outputs, a large number of random samples may be required to span the space. In this case, clustering techniques can be applied to limit memory requirements.

A straightforward example which demonstrates control of a plant represented by a second order linear system is presented here. An important advantage of GRNN is that it can treat nonlinear systems as well as linear systems, but for illustrative purposes a linear system will be used here. The plant has a natural period of six seconds and damping ratio of 0.5. The sample rate of the system is 2 Hz, so the outputs from the plant are available twelve times per cycle. For the first example, the desired output from the plant is a sine wave of unit amplitude and period six seconds.

The learning mode requires samples which span the space of anticipated control inputs and system outputs. One way to implement the learning mode with limited samples is to start with an initial control strategy that does reasonably well (perhaps linear control), and then have GRNN improve on that strategy. For this example, a 3-by-3 grid of points in the position-velocity phase space was selected to guarantee that the training space adequately represents the space over which the system operates.

The learning grid involves the origin (zero position and zero velocity) , and eight other points one unit away from the origin in position and/or velocity. Off-line calculations were performed to obtain the value of the control input which would cause the system to move (in two time steps) from each one of these nine points in the phase space to any of the same set of nine points. The nine-by-nine combination produces a set of 81 quintuples of numbers. Each set of numbers consists of the current state (position and velocity), the state desired in two time steps (position and velocity), and the value of the current control that gives the first step when the system transits from the current state to the desired state. Such a set can be used for the initial training set, and for learning in real time as the system evolves with time.

The position response in Figure 1 shows how the system evolves in time. The system starts at rest (the origin) and the control from the initial training data causes the trajectory to overshoot the desired sine wave slightly for the first two cycles. As the system evolves, it continues to pick up additional information, and quickly the system approaches the desired trajectory

In the second example, the plant and the initial training data are the same as previously, but the desired output is changed from a sinusoid of unit amplitude to a square wave of amplitude 0.5. and period 30 seconds. In this case, a different portion of the initial training space is reinforced (specifically, the portion devoted to maintaining a given value for some length of time). Figure 2 shows the control input required to produce a square wave is learned after the first few cycles. Furthermore, the neural network develops the capability to damp the oscillations that occur as a result of the step changes in position. These straightforward examples illustrate that with reasonable initial training, the neural network can adapt to varying system requirements.

ADAPTIVE NEURAL CONTROL AND FAULT TOLERANCE

Advances in the theory and technology of artificial neural networks provide the potential for new approaches to the problems of control, identification, and diagnosis for large complex space systems. The resulting control system can be augmented though use of various aspects of rule-based expert systems and fuzzy logic. The following describes an approach for an autonomous adaptive neural controller that can be used for performance improvement, health monitoring, and fault tolerance. To lend verisimilitude to the explanation, it will be assumed that a structural sub-system, such as a segmented mirror, is being controlled.

An important advantage of neural networks is the tremendous computational speed advantage achieved by massively parallel analog hardware. Therefore, it is proposed that the control system will use a high speed computational engine that is similar to the Lockheed Programmable Analog Neural Processor [6] (with the acronym PANP). The analog processor will be directed by a high level neural decision element (with the acronym HILENDEL). The General Regression Neural Network [5,12] (with the acronym GRNN) will aid in decision making.

The hierarchical architecture of the controller is shown in Figure 3 where the high level neural decision element (HILNDEL) directs the flexible, high speed Programmable Analog Neural Processor (PANP). The decision element continually monitors the system response and makes decisions about performance improvement, health monitoring, and fault tolerance. The decision element implements the required changes in the controller by down loading information to the PANP hardware. The PANP uses this information for high speed computation equivalent to the Kalman filter state estimation and feedback gain calculations, typical of most multi-input, multi-output control systems. The update is accomplished is such a way that the control loop is not interrupted, and as a result, the control algorithm adapts over time.

Since the physical parameters of space-based structures are likely to change over time, the decision element autonomously monitors the health of the system. If, while monitoring the system, there is a potential fault indication, the decision element identifies the particular fault condition. After the neural decision element recognizes the fault condition, it determines a specific response, based on stored responses for each known fault condition.

Specific responses for each fault condition consist of reconfiguring the PANP control system matrices. The method of reconfiguration is for the decision element to down-load an appropriate system description for an identified fault. After the PANP has been reconfigured, the decision element returns to the mode of evaluating performance and periodically updates both the system state estimation parameters and the feedback gain coefficients, as needed.

A distinguishing feature of the architecture is the Programmable Analog Neural Processor which is used as a high speed, flexible computation engine. The PANP receives inputs from the sensors, performs computations, and delivers outputs to the actuators. The block diagram in Figure 4 shows how the Programmable Analog Neural Processor could implement the Kalman Filter and the control law calculations. The structural system model is represented in the figure by three matrices (sometimes called A, B, and C in control terminology). The three matrices are termed the transition matrix, the control matrix, and the sensor matrix, respectively. Two additional matrices, which are determined by calculations from the decision element are the Kalman gain and the feedback control gain. The delay elements are implemented as analog delays, and are used to store past values of the state.

The past state values in combination with the actuator commands and sensor measurements are used to estimate the current state description of the structure. The actuator commands are calculated as the weighted sum of the estimated states. These feedback control gains (weights) are updated by the decision element to improve the measured system performance. The simulation of the PANP for purposes of the conceptual design is similar to what is shown in the figure. To minimize hardware requirements, the actual hardware design may be somewhat different, because many of the matrices will be sparse and terms will be combined to reduce the extent and complexity of the hardware.

Performance Improvement

Improving performance without interrupting the high speed control loop is an important goal. The high level neural decision element (HILNDEL) continually determines performance by evaluating line-of-sight (LOS) and structural vibrations. It calculates the desired control system matrices based on the system response, and periodically down-loads the updated matrices to the Programmable Analog Neural Processor (PANP).

The decision element evaluates all the sensors to determine the existence of spurious structural vibrations. If the vibrations are smaller than some threshold, the control system matrices are left unchanged. If the vibrations are moderately large, the decision element determines which actuators are most effective at suppressing the residual vibrations. The GRNN neural network is used to determine which control law coefficients or combination of coefficients should be changed to eliminate the vibrations. After the desired changes to the control law are determined, the matrices in the PANP are updated appropriately, and the decision element continues to evaluate the system response.

For illustrative purposes, simulation results are presented for a single segment of a six segment mirror. The Advanced Structures/Control Integrated Experiment (ASCIE) is a six segment mirror (developed under Lockheed Independent Research) used as a hardware test bed to evaluate significant control structure interaction. The goal is to implement alignment control (in piston, tilt, and petal) of a six segment mirror using 24 edge sensor measurements and 18 actuators. The two time domain plots in Figure 5 show step responses from a computer simulation of a single segment. The single segment model contains six sensor ,three actuator, four global system modes (eight states), non-proportional damping, and a strong coupling between the control hardware and structural support members (all of which are characteristic of large, flexible, space structures).

The left graph in Figure 5 shows how a step command in piston causes structural vibrations which are slow to damp if the controller is not tuned exactly to the structural model. In particular, there is a 10% discrepancy in modal frequency and a 50% discrepancy in modal damping of the fundamental mode. However, when the proper system model and control parameters are identified, the much improved response shown in the right graph is achieved. This example illustrates one technique to implement performance enhancement .

The Least-Mean-Square (LMS) algorithm can be used to adjust control system matrices to reduce structural vibrations. This algorithm continually modifies the control coefficients stored in the decision element using a gradient descent procedure to minimize a cost functional made up of line-of-sight (LOS) and structural vibration measures. Alternative tuning algorithms which can be evaluated include accelerated gradient and Newton methods (second order), and scalar descent methods (varying coefficients one at a time). Partition of the control gain matrices is a possible method to speed up convergence.

When the calculated control coefficients have reached a value sufficiently different than the current values, the decision element down loads the new coefficients to the PANP. Thus, the decision element continually recalculates the coefficients, but does not down-load them until the changes have reached some significantly different value. If there is not sufficient change, the control matrices are not updated. This procedure prevents spurious recycling.

Health Monitoring

Another important goal is to implement health monitoring which allows the system to recognize long term degradation and immediate faults. The high level neural decision element autonomously monitors the health of the system by evaluating line-of-sight and structural vibration measurements. If there is an indication that a potential fault exists, the decision element identifies the particular fault condition, and determines a specific response, based on stored data.

Health monitoring must discriminate between long term degradation and immediate failures, and must respond rapidly in the case of a failure. One method of validating fault tolerance with an experiment is to remove various critical structural members and/or actuators to observe the time of re-convergence and the resulting level of performance. The neural network GRNN can be used to discriminate between failure candidates. The following explanation describes how a typical health monitoring algorithm deals differently with structural member removal and sensor or actuator failure.

Typical spacecraft have direct redundancy of sensors. Therefore, the most straightforward example of health monitoring considers sensor failure. The redundancy allows estimation of the measurements at each sensor based on other sensor measurements. If the difference between the actual and the estimated measurements increases slowly over time, it may mean gradual degradation. If there is rapid short term increase in the difference for a single sensor, it can mean there is a sensor failure.

To identify a potential actuator failure, the predicted sensor measurements are compared with the actual system measurements. If there are significant differences, these differences are input to the GRNN neural decision element. GRNN is used to discriminate which actuator has failed. If the probability exceeds some threshold, then fault accommodation is initiated.

Because the structural sensitivity matrix is not readily available when a structural member is removed or damaged, the discrimination situation is more difficult. One approach is to develop a separate structural model and sensitivity matrix for each structural member which can be removed or damaged. Complexity can be reduced somewhat because the structural members are typically geometrically similar and because there are symmetries in the typical space structure. As before, the GRNN neural network is used to determine which structural element has been removed. Alternatively, GRNN might determine the modified system response without actually identifying the specific structural member that has been removed.

An example of the capability of neural-based health monitoring is presented in Figure 6. Data shown in these two time domain plots were generated by performing a computer simulation with a single segment of the ASCIE six segment mirror described previously. The left graph illustrates the step response with the optimized control law. The right graph shows the degraded performance resulting from a simulated structural fault. For this example, the signature of the fault is an oscillatory response to the step input along with a small steady state error. GRNN determines which particular fault has occurred. Once the fault is identified and the associated reconfiguration is determined, fault tolerance is initiated.

Fault Tolerance

Fault tolerance is another goal for a neural controller which will allow the system to recover autonomously from assorted hardware faults. Specific responses are stored in the decision element for each specific fault condition. After the fault condition has been determined, the decision element immediately reconfigures the PANP by down loading a new set of control system matrices. The GRNN neural network is used both to determine the probability of various candidate faults and to determine the appropriate corresponding reconfiguration.

Fault tolerance is implemented by down-loading the appropriate stored reconfiguration to the PANP. After the new configuration is implemented in the PANP, the decision element returns to monitoring performance and periodically updating the PANP coefficients. Since the performance with the reconfigured system may be worse than the nominal performance, the threshold for down-loading control law modifications may require adjustment.

An example of the capability to implement fault tolerance is illustrated in Figure 7 by the computer simulation results on a model of a single segment of the ASCIE six segment mirror. The left graph shows the degraded system response to a step input due to a structural fault. The health monitoring system recognizes that a fault has occurred and identifies the failed component. The GRNN algorithm then determines the appropriate reconfiguration based on stored responses for specific failures. In the example, the system controller is reconfigured based on the assumed structural faults. The return to a nearly optimal system is shown in the right graph with less high frequency vibration and no steady state error. This illustrates one approach to fault accommodation.

CONCLUSION

This paper has presented some illustrations of the use of neural networks for control, identification, and diagnosis. Neural networks are shown to be part of a hierarchy of intelligent control where a high order decision element monitors and supervises lower order elements for sensing and actuation. An important advantage of neural networks is the tremendous computational speed advantage achieved by massively parallel analog hardware. A first generation example is the Programmable Analog Neural Processor (PANP) which can be used as a flexible high speed computation engine. The coefficients for the control law can be adjusted on the fly in PANP without interrupting the control loop. The General Regression Neural Network (GRNN) or the Probabilistic Neural Network (PNN) are candidates to aid in high level decision making.

The challenges inherent in any control application include modeling, simulation, control design, health monitoring, fault tolerance, hardware implementation and experimental verification. In order to validate these approaches for space-based missions, it is necessary to have ground-based experimental demonstrations which integrate neural control and modern control and which integrate analog neural hardware with conventional hardware. There is still a long way to go, but the challenge is there, and the pay-off is promising.

tREFERENCES

1. K.S.Narendra and K.Parathasarathy, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. on Neural Networks, March 1990.

2. D.E.Rumelhart, J.L.McClelland and the PDP Research Group,Parallel Distributed Processing, Vol.1: Foundations, Ch.8, MIT Press,Cambridge MA.1986

3. D.F.Specht, Probabilistic Neural Networks for Classification, Mapping or Associative Memory, Proc. IEEE International Conference on Neural Networks, June 1988.

4. D.F.Specht, Probabilistic Neural Networks, Neural Networks Journal, Vol. 3, 1990.

5. D.F.Specht, General Regression Neural Networks, IEEE Trans. on Neural Networks, November 1991.

6. W. A. Fisher, R. J. Fujimoto, and R. C. Smithson, A Programmable Analog Neural Network Processor, IEEE Trans. on Neural Networks, March 1991

7. Special Issue on Neural Networks I: Theory and Modeling, Proc. IEEE, Sept. 1990.

8. Special Issue on Neural Networks II: Analysis, Techniques and Applications, Proc. IEEE, Oct. 1990.

9. Special Issue on Neural Networks in Control Systems, IEEE Control Systems Magazine, April 1990.

10. Special Issue on Neural Networks in Control Systems, IEEE Control Systems Magazine, April 1992.

11. E. Sanchez-Sinencio and C. Lau, Artificial Neural Networks, IEEE Press, New York,1992.

12. D.B.Schaechter and H.E.Rauch, Application of Neural Networks to Control Systems, Proc. AIAA /AAS Astrodynamics Conference, August 10-12, 1992.

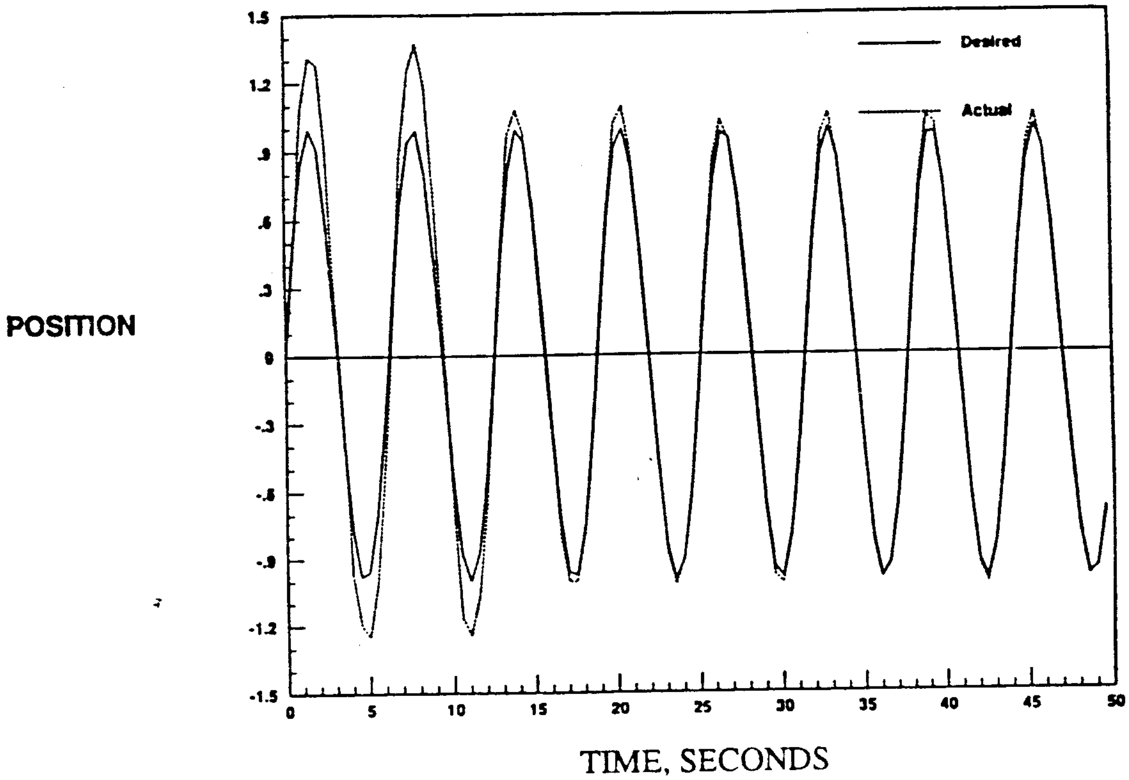

Fig. 1. Plot of position time history shows the neural controller causes the system output to converge to the desired sine wave after first few cycles

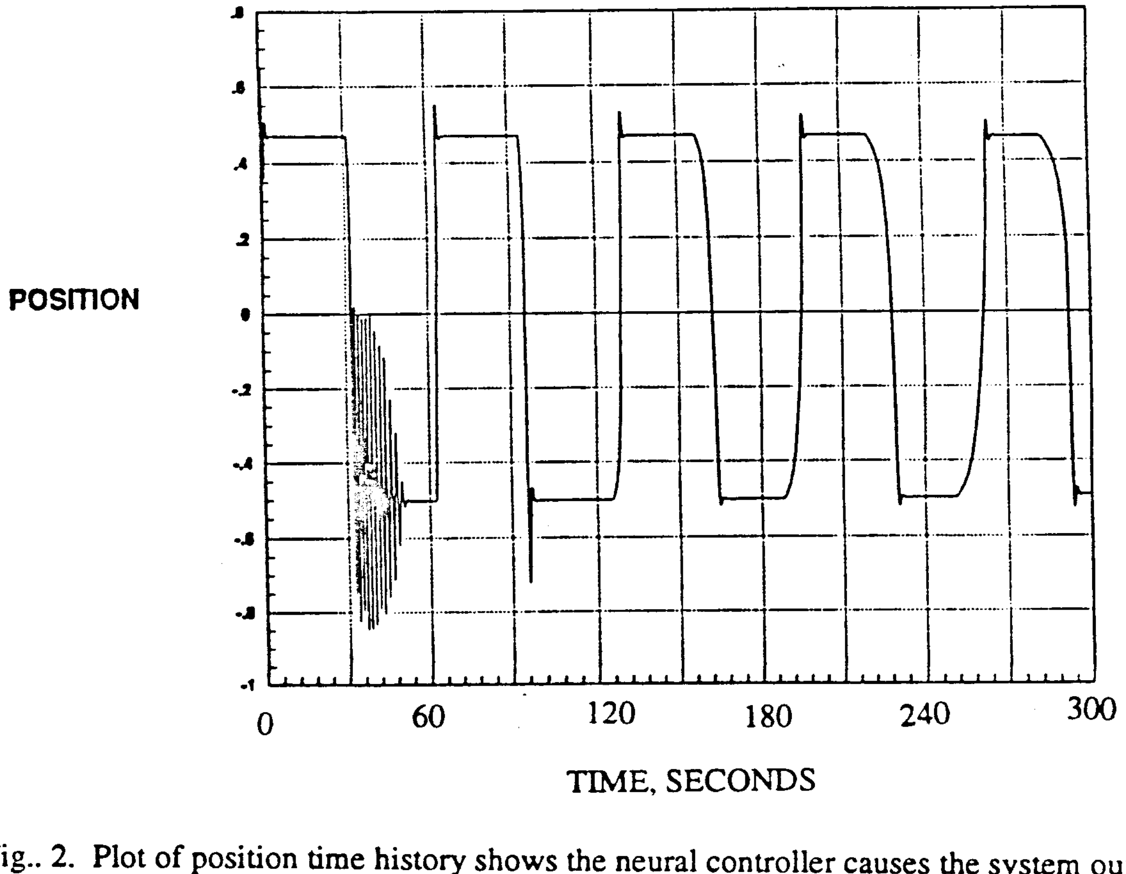

Fig.. 2. Plot of position time history shows the neural controller causes the system output to converge to the desired square wave after first few cycles

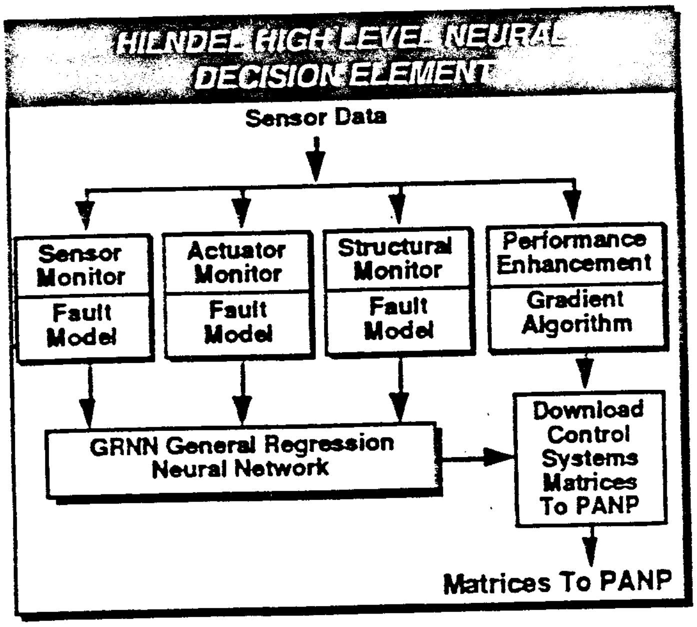

Fig. 3. High level neural decision element (HILENDEL) uses General Regression Neural Network (GRNN) to direct Programmable Analog Neural Processor (PANP) for performance enhancement, health monitoring, and fault tolerance.

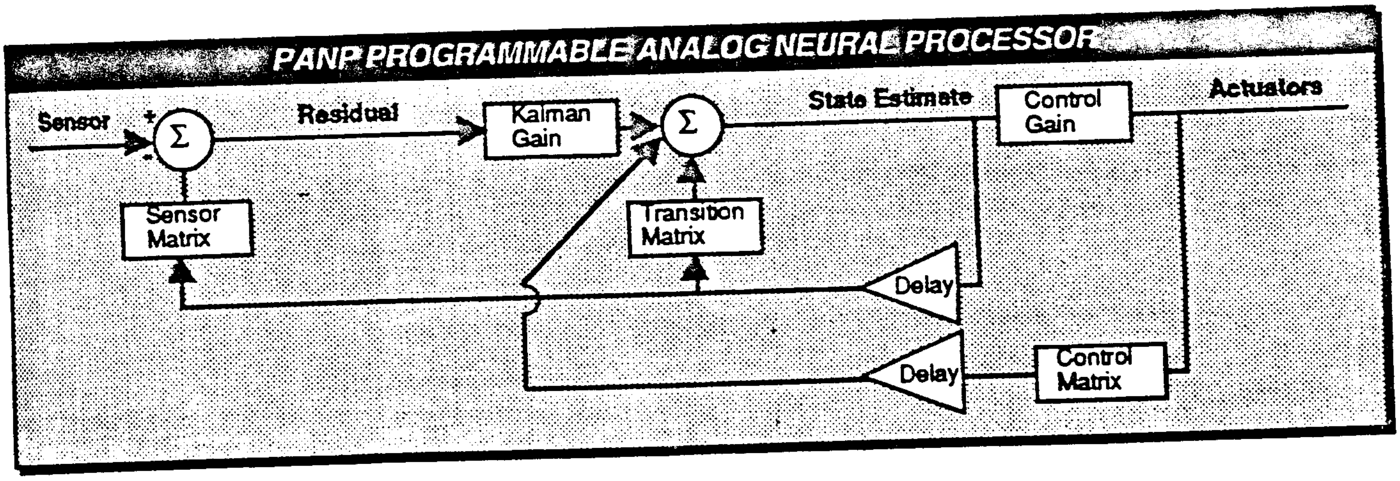

Fig. 4. Block diagram of Programmable Analog Neural Processor (PANP) which implements high performance control loop with Kalman filter and control gain.

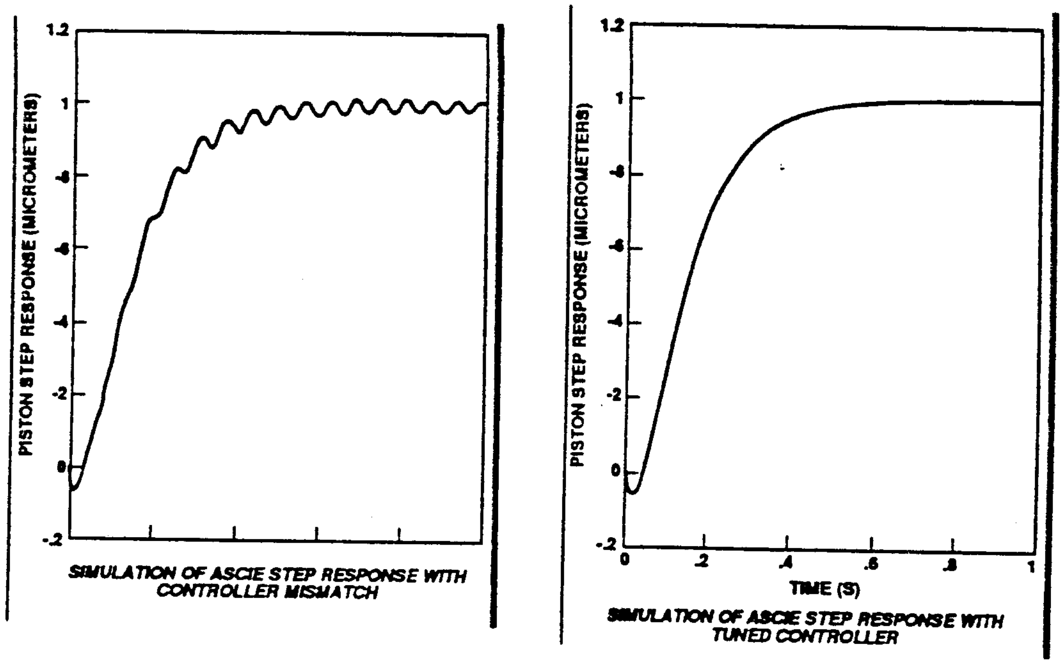

Fig. 5. Simulation of step response for one segment of six segment mirror for the Advanced Structures/Control Integrated Experiment (ASCIE) illustrates performance enhancement. Step response on left shows structural vibrations which are slow to die out because controller is not tuned exactly to structural model. Improved step response on right shows results from tuning controller gains.

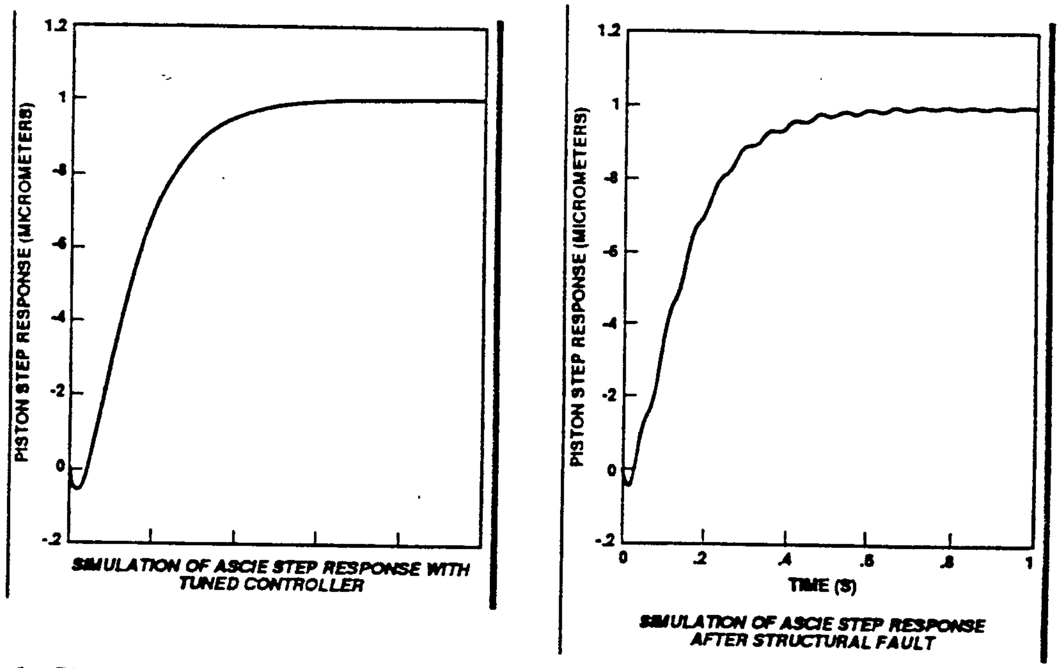

Fig. 6. Simulation of step response for one segment of six segment mirror for the Advanced Structures/Control Integrated Experiment (ASCIE) illustrates health monitoring. Step response on left shows results with tuned controller gains. Step response on right shows structural vibrations and steady state bias due to simulated structural fault.

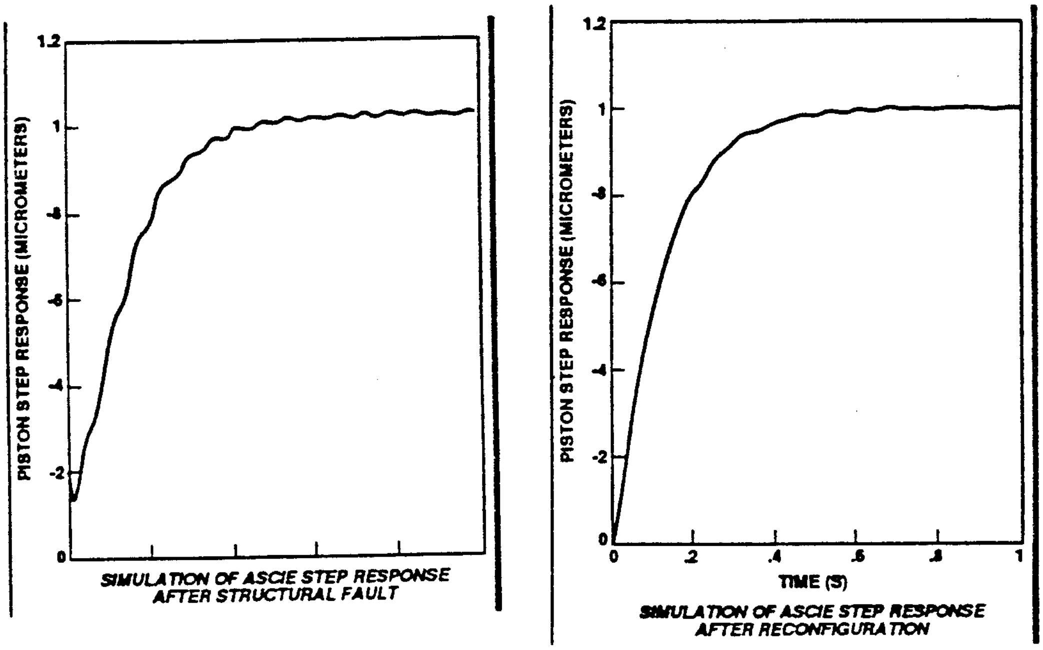

Fig. 7. Simulation of step response for one segment of six segment mirror for the Advanced Structures/Control Integrated Experiment (ASCIE) illustrates performance enhancement. Step response on left shows structural vibrations due to simulated structural fault. After appropriate system reconfiguration, the improved step response on right has less vibration and no steady state error gains.