

110 00
127124
P-12

NASA Technical Memorandum 105898

Analysis of Fault-Tolerant Neurocontrol Architectures

T. Troudet
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

and

W. Merrill
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

Prepared for the
31st Conference on Decision and Control
sponsored by the Institute of Electrical and Electronics Engineers
Tucson, Arizona, December 16-18, 1992



(NASA-TM-105898) ANALYSIS OF
FAULT-TOLERANT NEUROCONTROL
ARCHITECTURES (NASA) 12 p

N93-12305

Unclass

G3/08 0127124



ANALYSIS OF FAULT-TOLERANT NEUROCONTROL ARCHITECTURES.

T. TROUDET ¹ and W. MERRILL
Advanced Control Technology Branch
NASA Lewis Research Center
Cleveland, Ohio 44135

Abstract

The fault-tolerance of analog parallel distributed implementations of a multivariable aircraft neurocontroller is analyzed by simulating weight and neuron failures in a simplified scheme of analog processing based on the functional architecture of the ETANN chip. The neural information processing is found to be only *partially* distributed throughout the set of weights of the neurocontroller synthesized with the backpropagation algorithm. Although the degree of distribution of the neural processing, and consequently the fault-tolerance of the neurocontroller, could be enhanced using *Locally Distributed Weight and Neuron Approaches*, a satisfactory level of fault-tolerance could only be obtained by retraining the degraded VLSI neurocontroller. The possibility of maintaining neurocontrol performance and stability in the presence of single weight or neuron failures was demonstrated through an *automated retraining procedure* of the neurocontroller based on a pre-programmed choice and sequence of the training parameters.

1. Introduction.

Recent advances in the domain of *analog* VLSI [1-3] have started to unveil the benefits of massively parallel distributed implementations of neural networks. When weights and neurons are implemented as separate physical entities, the neural information can be simultaneously processed by the weights and neurons of the same layer in the case of feedforward neural networks, or by all the weights and neurons in the case of fully interconnected neural networks. Such a massive parallelism provides VLSI implementations of neural networks with processing speeds that are unmatched by ordinary computing means. When the "synaptic" weights are implemented as analog devices as in Ref.[2], weight degradations due to time-decay, or weight failures due to anomalous charge leakages, may however prevent from achieving the desired neuroprocessing. Due to their distributed structure, analog VLSI neural networks have the *potential* to be inherently fault-tolerant with respect to such degradations/failures, since the degradation or failure of a weight or neuron does not alter the neural processing by the many other weights or neurons operating in parallel. At the same time, cross-correlations of the signal incurred through the multiple successive layers of feedforward networks, or through the feedback interconnections of fully interconnected networks, have the potential to alter the outcome of the overall neural processing in the presence of a single weight or neuron failure. Moreover, analog VLSI implementations are expected to process neural information with a lesser precision than the accuracy usually obtained with conventional digital computing architectures, due to the analog nature of the signal.

The objective of this paper is to address major issues pertaining to the performance and the fault-tolerance of *analog* VLSI neural networks in the context of control applications. Towards this objective, simulation results are presented for potential analog VLSI implementations of a robust dynamic neurocontroller that was previously designed for a multivariable aircraft control problem [4]. In this work, the term *fault-tolerance* is defined as maintaining neurocontrol performance and stability in the presence of hardware *failure(s)* or component *degradation(s)* of the analog *VLSI* implementation of the neurocontroller, whereas the term *robustness* is defined as maintaining neurocontrol performance and stability in the presence of *uncertainties* associated with the *modelling process* [4].

The paper is organized as follows. The functional structure of the neurocontroller is briefly reviewed in Section 2 in relation to the multivariable aircraft control problem example. The computational structure of the neurocontroller is analyzed in Section 3, with special emphasis given to estimate the degree of distribution of the neural processing among the various weights and neurons. In Section 4, the ability of an all analog VLSI implementation to achieve the desired neurocontrol performance is analyzed by simulating simplified schemes of analog processing based on the functional architecture of the ETANN chip [2-3].

¹T. Troudet is with Sverdrup Technology, Inc., 2001 Aerospace Parkway, Brook Park, Ohio 44142.

2. Functional Structure of the Neurocontroller.

The neural network chosen for this analysis is the dynamic neurocontroller that was designed in Ref.[4] to provide independent control of pitch rate and airspeed responses to pilot command inputs for an integrated airframe/propulsion longitudinal dynamics model of a modern fighter aircraft. The characteristics of the neurocontrol are best illustrated through the functional blocks of the closed-loop evaluation architecture of Fig.1.

In this two control inputs - two control outputs example, the task is that of following the trajectories generated from a linear model of the desired vehicle response dynamics to pilot command inputs [4, 5]:

$$\bar{z}_c = f_{filter}(\bar{z}_{SEL}); \quad (1)$$

with $\bar{z}_{SEL} = [V_{SEL}, Q_{SEL}]^T$, where V_{SEL} is the pilot velocity command in ft/s, and Q_{SEL} is the pilot longitudinal stick deflection in inches; $\bar{z}_c = [V_c, Q_c]^T$, where the subscript "c" refers to the ideal response in V and Q with units of ft/s and deg/s respectively; and f_{filter} is a linear function given in Refs.[4, 5]. The vehicle outputs to be controlled are

$$\bar{z} = [V, Q]^T; \quad (2)$$

where V is the aircraft velocity in ft/sec, and Q is the pitch rate in deg/s.

As indicated in Fig.1, the inputs to the neurocontroller are sampled values of the *measured vehicle outputs* $\bar{z}(t)$, the *tracking errors* $\bar{e}_z(t) = \bar{z}_c(t) - \bar{z}(t)$, and the *estimated* values of the *state vector* of the vehicle, $\hat{\bar{x}}(t)$. Estimates $\hat{\bar{x}}(t)$ of the state vector are solutions of the linearized dynamics equations of the vehicle model:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}_a, \quad \bar{z} = C\bar{x}; \quad (3)$$

where A , B , and C are the *nominal* system matrices listed in Ref.[5]. In Eq.(3), $\bar{x}(t)$ is the 9-component vector:

$$\bar{x} = [u, w, Q, \theta, h, N2, N25, P6, T41B]^T; \quad (4)$$

u being the aircraft body axis forward velocity (ft/sec), w the aircraft body axis vertical velocity (ft/sec), Q the aircraft pitch rate (rads/sec), θ the pitch angle (rads), h the altitude (ft), $N2$ the engine fan speed (rpm), $N25$ the core compressor speed (rpm), $P6$ the engine mixing plane pressure (psia), and $T41B$ the engine high pressure turbine blade temperature ($^{\circ}$ R).

The vehicle model consists of an integrated state-space representation for a modern fighter aircraft powered by a two-spool turbofan engine and equipped with a two-dimensional thrust-vectoring and reversing nozzle. For this model, the control input vector \bar{u}_a , in Eq.(3), is

$$\bar{u}_a = [WF, \delta TV]^T; \quad (5)$$

WF being the engine main burner fuel flow rate (lbm/hr), and δTV the nozzle thrust vectoring angle (deg). The fuel flow actuator is modelled as the second order system

$$G_{WF}(s) = 500/(s^2 + 60s + 500); \quad (6)$$

with a maximum fuel flow rate $|WF|_{max} = 10,000 \text{ lbm/hr}$ (maximum deviation from the trim value), and a rate limit $|\dot{WF}|_{max} = 20,000 \text{ lbm/hr/s}$. The thrust vectoring actuator is modelled as the first order system

$$G_{\delta TV}(s) = 15/(s + 15); \quad (7)$$

with a maximum thrust vector angle $|\delta TV|_{max} = 10 \text{ deg}$ (maximum deviation from the trim value), and a rate limit $|\dot{\delta TV}|_{max} = 20 \text{ deg/s}$. The response of the non-linear actuators, $\bar{u}_a(t)$, to the output of the dynamic neurocontroller, $\bar{u}_c(t)$, changes the state of the vehicle model to achieve the desired tracking performance (Fig.1).

The neurocontroller was trained in Ref.[4] to track pilot commanded trajectories that were generated as follows. The pilot selected pitch rate was a *doublet* centered at a time t_c between 2.5s and 5s, with the characteristics: $Q_{SEL}(t) = Q_0$ for $t \leq t_c$; $Q_{SEL}(t) = -Q_0$ for $2t_c \geq t > t_c$; $Q_{SEL}(t) = 0$ for $t > 2t_c$. The pilot selected airframe velocity was a step function characterized by $V_{SEL}(t) = 0$ for $t \leq 0$, and $V_{SEL}(t) = V_0$ for $t > 0$. The maximum intensities $|Q_0|$ and $|V_0|$ of the randomly selected input commands were bounded

by $Q_{max} = 0.5in$ and $V_{max} = 20ft/sec$. These commanded inputs $Q_{SEL}(t)$ and $V_{SEL}(t)$, which represent the frequency-content of typical pilot command inputs, were subsequently filtered over a period of 12s to generate the commanded trajectories $Q_c(t)$ and $V_c(t)$ [4]. Once the neural network had learned to track such commanded trajectories, the neurocontroller was found to exhibit good robustness through stability margins in phase and vehicle output gains, and maintained performance and stability in the presence of error loop failures. A more detailed evaluation of the neurocontroller performance can be found in Ref.[4].

3. Computational Structure of the Neurocontroller.

The neurocontroller of Fig.1 is a feedforward neural network with 13 input units (2 inputs for the measured outputs \bar{z} , 2 inputs for the tracking errors \bar{e}_x , and 9 inputs for the estimate \hat{x} of the vehicle state vector), 10 neurons in the first hidden layer, 10 neurons in the second hidden layer, and 2 neurons in the output layer, the WF -neuron and the δTV -neuron respectively. The neuron thresholds are provided by a unit whose output is permanently set to +1, and which is synaptically connected to all the neurons. For notation convenience, this unit, referred to as the *bias* unit, is placed in the input layer. Neuron i of the p^{th} layer is represented by $n_{p,i}$, and its output by $o_{p,i}$. A weight connecting neuron $n_{p,i}$ to neuron $n_{p+1,j}$ is represented by $w_{p+1,j;p,i}$. Each neuron has the sigmoid activation function

$$output = \tanh(input) , \quad (8)$$

which limits its output to the interval $[-1, +1]$ for any input signal. The input units simply fan-out the input data values to the neurons of the first hidden layer. The outputs of the WF -neuron and the δTV -neuron in the output layer are the components of the normalized commanded control input vector

$$\bar{u}_c^s(t) = [WF_c(t)/|WF|_{max}, \delta TV_c(t)/|\delta TV|_{max}]^T , \quad (9)$$

which is applied to the actuators as indicated in Fig.1.

To estimate the degree of distribution of the neural information processing through the set of weights of the neurocontroller, closed-loop neurocontrol performance was evaluated for every network configuration where a *single weight* or a *single neuron* is removed. This was achieved through Monte Carlo simulations of closed-loop pitch-rate/velocity responses to pilot command inputs, after setting the value of a weight, or forcing the output of a neuron, to zero during the entire neural computation. The simulation results as reported in Fig.2 indicate which weights and which neurons have the most significant effect on the neurocontroller performance.

Individual Weight Contributions to the Neural Computation.

Among the weights connecting the input layer to the first hidden layer, $w_{2,2;1,13}$ is found to lead to unstable pitch rate and velocity responses when it is set to zero. When $w_{2,2;1,13} = 0$, both closed-loop responses exhibit growing oscillations that are particularly pronounced in velocity-tracking. Such a critical interconnection is represented by a bold line in Fig.2 to indicate the loss of stability of the neurocontroller when the interconnection is removed from the network. When $w_{2,4;1,4} = 0$, or $w_{2,5;1,2} = 0$, or $w_{2,5;1,7} = 0$, the closed-loop responses present an offset in steady-state tracking, and the corresponding interconnections are represented by dashed lines in Fig.2. The other weights of the input layer individually have little effect on the neurocontroller performance, and are represented by thin lines in Fig.2.

To further estimate the influence of the critical interconnection $w_{2,2;1,13}$ on the neurocontroller performance, closed-loop system responses were simulated for the value of $w_{2,2;1,13}$ that is a fraction of the original value $w_{2,2;1,13}^{(0)}$ obtained from backpropagation training [4]. This fractional weight value was chosen of the type $w_{2,2;1,13}/w_{2,2;1,13}^{(0)} = k/(k+1)$. Closed-loop simulations indicated that the unstable oscillations in pitch-rate and velocity responses disappear if $w_{2,2;1,13}/w_{2,2;1,13}^{(0)} \geq 1/2$, for which values the neurocontroller performance is found to be close-to-nominal. In anticipation of the analog implementation of Section 4, the latter result suggests the possibility of enhancing the degree of distribution of the neural processing among the weights of the input layer by substituting $w_{2,2;1,13}$ with two parallel weights of magnitude $w_{2,2;1,13}^{(0)}/2$. This method referred to as the *Locally Distributed Weight Approach (LDWA^(k))* is illustrated in Fig.3. Likewise, applying $LDWA^{(k=2)}$ to the Q -input gains $w_{2,5;1,2}$ and $w_{2,5;1,7}$, and to the e_Q -input gain $w_{2,4;1,4}$ would enhance the degree of distribution of the neural processing in the first layer. This would also provide a close-to-nominal performance when a single interconnection of the input layer is removed from the network.

Among the weights interconnecting the two hidden layers, $w_{3,1;2,5}$ and $w_{3,1;2,4}$ are essential for maintaining neurocontrol performance and stability. Although less critical, each of the six interconnections

represented by dashed lines in Fig.2 contributes noticeably to the neural computation for maintaining neurocontrol performance. As above, selective application of LDWA to these interconnections would enhance the distribution of the neural processing, and would maintain neurocontrol performance and stability when a hidden weight is removed from the network.

Among the interconnections of the output layer, $w_{4,2;3,1}$ and $w_{4,2;3,6}$ are essential for maintaining neurocontrol performance and stability. This is illustrated for the most critical interconnection $w_{4,2;3,1}$ by the closed-loop responses in Fig.5, where the input command consists of the pitch rate pulse $Q_{SEL}(t) = 0.5\text{in}$ for $t \leq 3\text{sec}$, $Q_{SEL}(t) = 0$ for $t > 3\text{sec}$, applied simultaneously with the velocity step command $V_{SEL}(t > 0) = 20\text{ft/sec}$. It is noted that this input command has a different frequency content than that of the signal used to train the neurocontroller. To estimate the effect of time-delays on the neurocontroller performance, closed-loop responses were simulated in Fig.1 by introducing an additional 50ms delay in both neurocontrol channels, at location α in Fig.1, for a 5ms sampling time and a delay $\Delta = 5\text{ms}$ at the output of the neural state estimator.

As shown in Fig.5, both pitch-rate and velocity responses are highly unstable if $w_{4,2;3,1} = 0$. Control stability is however regained if $w_{4,2;3,1}/w_{4,2;3,1}^{(0)} \geq 2/3$. The same observations apply to $w_{4,2;3,6}$. Although less critical, the interconnection $w_{4,2;3,4}$ represented by a dashed line in Fig.2 contributes noticeably to the neural computation for maintaining neurocontrol performance. It could also be distributed via LDWA to enhance the fault-tolerance of the neurocontroller to the loss of a single output weight. All other interconnections of the output layer represented by thin lines in Fig.2 have little individual effect on the neural computation.

To further estimate the extent to which the neuro-processing is distributed among the non-critical interconnections (represented by thin lines in Fig.2), the neurocontroller performance was evaluated after removing multiple weights connecting into the WF-neuron of the output layer. Individual sets of 50 network configurations were randomly generated by simultaneously removing n interconnections to the WF-neuron. Only from $n = 5$ did some closed-loop responses start to deviate significantly from the nominal responses. As the number of removed interconnections increased from $n = 1$ to $n = 4$, the neurocontroller performance was found in all cases to degrade gracefully within acceptable margins of the nominal responses.

The neuron thresholds of the neurocontroller trained by backpropagation in Ref.[4] have very small magnitudes, and the neurocontroller performance remains unchanged if they are set to zero. This indicates that these network parameters were not used by the backpropagation algorithm to provide the desired neurocontrol performance, and that they could have been eliminated before training. As will be seen in the next section, these parameters will however be used by the backpropagation algorithm to rearrange the weights of the neurocontroller for maintaining control performance and stability in the presence of weight degradation or neuron failure.

The simulation results also seem to point to some correlation between the magnitudes of the weights and their individual contributions to the neural computation. For example, the critical weights $w_{4,2;3,1}$ and $w_{4,2;3,6}$ which are essential for maintaining neurocontrol stability have the largest absolute magnitudes among the weights connecting into the δTV -neuron, (five times and twice the magnitude of the next largest weight, respectively). Among the weights connecting into neuron $n_{3,1}$, the absolute magnitude of the critical weight $w_{3,1;2,5}$ is also among the largest ones. Yet, weights of large magnitude are not necessarily essential to the stability of the neurocontroller, as is the case for the weights connecting into the WF-neuron. Although the weights connecting into the WF-neuron have much larger absolute magnitudes than those connecting into the δTV -neuron, the removal of any such single weight still allows for satisfactory command tracking. This observation suggests the possibility of truncating the weights of largest absolute magnitudes without altering the quality of the neurocontrol, and thereby enhance the performance of neurocontrol implementations by reducing the adverse effect that weight uncertainties have on the analog processing.

Individual Neuron Contributions to the Neural Computation.

Among the neurons of the first hidden layer, $n_{2,2}$, $n_{2,4}$ and $n_{2,5}$ lead to unstable closed-loop responses when their output is forced to zero during the neural computation. As expected, these critical neurons (highlighted in bold dashed lines in Fig.2) are also connected to critical interconnections (bold lines in Fig.2). For $o_{2,2} = 0$, or $o_{2,4} = 0$, or $o_{2,5} = 0$, the system responses are similar to those for $w_{2,2;1,13} = 0$, or $w_{3,1;2,4} = 0$ or $w_{3,1;2,5} = 0$ respectively, yet with larger deviations from the ideal responses. To further estimate the influence of these critical neurons on the neurocontroller performance, closed-loop system responses were simulated for neuron outputs having fractional values $k/(k+1)$ of the nominal outputs. The simulation results indicated that the neurocontroller is stable if the output of $n_{2,2}$ is $2/3$ of its nominal value, and if the

outputs of $n_{2,4}$ and $n_{2,5}$ are 3/4 of their nominal values.

In anticipation of the analog implementation of Section 4, the latter result also suggests the possibility of enhancing the degree of distribution of the neural processing among the neurons of the first hidden layer through the *Locally Distributed Neuron Approach* ($LDNA^{(k)}$) illustrated in Fig.4. A single neuron is replaced by k neurons whose output voltages are fed into separate networks of interconnections. These networks of interconnections have the same configuration as the original network of interconnections, but the weight magnitudes are $1/k$ times the nominal values.

For $o_{2,3} = 0$, the pitch rate and velocity responses exhibit a limit cycle behavior. This limit cycle is removed by applying $LDNA^{(k=2)}$ to neuron $n_{2,3}$ represented in thin dashed lines in Fig.2. Although $o_{2,6} = 0$ leads to a significant offset in steady-state tracking, the neurocontroller performance is close-to-nominal if $LDNA^{(k=2)}$ is applied to neuron $n_{2,6}$ represented in thin dashed lines in Fig.2. All the other neurons of the first hidden layer individually contribute little to the neural computation.

Among the neurons of the second hidden layer, $n_{3,1}$ and $n_{3,6}$ represented in bold dashed lines in Fig.2 lead to unstable closed-loop responses when their output is forced to zero. The system responses for $o_{3,1} = 0$ or $o_{3,6} = 0$ are very similar to those for $w_{4,2;3,1} = 0$ or $w_{4,2;3,6} = 0$, respectively. Close-to-nominal neurocontrol performance is still obtained if the outputs of $n_{3,1}$ and $n_{3,6}$ are 2/3 and 3/4 of their respective nominal values. Although less critical, neuron $n_{3,4}$ represented in thin dashed lines in Fig.2 contributes noticeably to the neural computation for maintaining neurocontrol performance. Close-to-nominal neurocontrol performance can yet be achieved by applying $LDNA^{(k=2)}$ to $n_{3,4}$. All other neurons of the second hidden layer have little individual effect on the neural computation.

To further estimate to which extent these non-critical neurons contribute to the neurocontrol, the neurocontroller performance was evaluated after removing $n \geq 2$ non-critical neurons from the second hidden layer. For $n = 2$, all the system responses tested showed little deviation from the ideal trajectories. For $n = 5$, nearly all system responses had a large average tracking error. The neurocontroller performance was found to degrade gracefully as n was increased from 2 to 5.

The influence diagram of Fig.2 clearly indicates the existence of predominant datapaths in terms of clusters of weights and neurons that are essential for maintaining neurocontrol performance and stability. In view of Figs. 3 & 4, Fig.2 shows that the degree of distribution of the neural processing can be enhanced through selective application of LDNA to the neurons and LDWA to the weights. This procedure is however costly in its increase of interconnections. It also only provides a limited increase in the fault-tolerance of the neurocontroller, since degradations or failures of practical implementations may result in situations where a neuron output voltage gets stuck at $o_{max} = +1$ or $o_{min} = -1$, or a weight deteriorates to $W_{max} = +1$ or $W_{min} = -1$. Alternatives, or complementary approaches, to enhance the fault-tolerance of the neurocontroller in the presence of such failures will therefore be proposed in the next section within the context of the ETANN chip.

4. Performance and Fault-Tolerance Estimations of Analog Parallel Distributed Implementations of the Neurocontroller.

In the Electrically Trainable Analog Neural Network (ETANN) [2-3], the analog input signals are first passed to the input layer which consists of a series of buffers with an active output range from 0 to 3.5 volts. These buffers are connected to the first hidden layer through synapse multiplier circuits. Each synapse multiplier circuit generates a differential current that is proportional to the *product* of its *input voltage*, e.g. a buffer output voltage, and its *differential weight voltage*, i.e. the weight. The output differential currents of these synapse multiplier circuits are then summed following Kirchoff's Law, and the final summation is input to a "neuron" of the first hidden layer. The "neurons" are designed as operational amplifiers with a variable gain, and an active output range from 0 to 3.2 volts. The neuron output voltages are subsequently input to the synapse multiplier circuits of the second hidden layer, and the analog signal processing proceeds likewise through the next layer, and on. The analog outputs of the neurons of the last layer are identified as the scaled/shifted values of the commanded control input vector given in Eq.(9).

The differential weight voltage of a synapse multiplier circuit is obtained as the difference in floating gate thresholds of a pair of EEPROM cells, the *weight cell* and the *reference cell*. Differential weight voltages are constrained between -2.5 volts and +2.5 volts. Weight values can be changed in strength and polarity by altering the charges of the floating gates. Such charge modifications are obtained through Fowler-Nordheim tunneling by applying variable voltage pulses to the floating gates. As a result, "weights" can only be stored with a limited accuracy due to hardware uncertainties. The possibility of implementing the neurocontrol

synthesized in Ref.[4] on such fully analog parallel distributed architectures therefore depends on the weight accuracy that can be tolerated by the neural computation for maintaining control performance and stability. To estimate that dependence, the performance of the neurocontroller was evaluated by simulating the ETANN function at typical operating conditions.

Performance. With the simplified scheme of the ETANN chip proposed in Ref.[3], the actual values of all weights and output voltages are scaled/shifted between -1 and +1, and the neuron transfer characteristics are simulated as:

$$\text{output} = \tanh(\text{gain} \times \text{total_input}) \quad (10)$$

where the total input signal is amplified with an adjustable gain factor.

The ETANN function was simulated with the activation function given in Eq.(10) for a gain factor of 1. The absolute maximum values of each of the thirteen inputs to the neurocontroller were first estimated through Monte Carlo simulations of closed-loop responses over a wide range of pilot input commands. These estimates were then used to rescale the thirteen input data to the neurocontroller, and the corresponding weights of the input layer. To simulate the input buffers of the ETANN chip, the fan-out input units described in Section 3 were replaced by linear-thresholding units with an output range of [-1, +1]. Since large weight values were found to be not necessarily essential to the stability of the neurocontroller, the possibility of truncating the weights without affecting the neurocontroller performance was analyzed through Monte Carlo simulations. In this application where the maximum actual weight value obtained in Ref.[4] is about 1.6, the neurocontroller performance is found to be unaffected up to a weight truncation of 1, i.e. for $-1 < w < 1$. Since analog implementations inherently have a limited resolution in the weights setting, truncating the actual weight values within [-1, +1] will actually provide a closer-to-nominal performance of the analog neurocontroller by reducing the effect of weight uncertainties on the neural computation.

The effect of small uncertainties in the hardware characteristics of the synapse electronic components was simulated by random fluctuations of the weights w around their target values w_t . For an equivalent n -bit precision in the weights setting, a weight w can take any arbitrary value between $w_t - 1/2^{n-2}$ and $w_t + 1/2^{n-2}$. Because of the stochastic nature of the weight setting, Monte Carlo simulations of closed-loop responses were run for many such random settings of the weights. The neurocontroller is found to be unstable for a weight precision less than or equal to 4 bits. For 5-bit and 6-bit precisions, closed-loop responses are stable but deviate substantially from the commanded trajectories during both transients and steady-states. Although the neurocontroller performance would be acceptable for 7-bit precision, close-to-nominal performance is achieved for and above 8-bit precision. The statistical distribution of the tracking errors is illustrated in the closed-loop responses of Fig.6 for 20 random settings of the weights with 8-bit precision.

It is worth mentioning that this simulation based on the neuron transfer characteristic of Eq.(10) does not take into account the internal dynamics of the ETANN chip, in particular the settling times of the operational amplifiers. Simulating the analog signal in the time domain would demand a numerical effort far beyond the scope of this work. It is however clear that the dynamics of the electronic components could influence the mode of operation of the neurocontroller. If the settling times of the op-amps are not small in comparison to the response time of the closed-loop system, it may be necessary to prevent the transient phase of the analog processing from altering the closed-loop control, for example by holding the analog signals at the input and output of the neurochip. However, if the settling times are small in comparison to the response time of the closed-loop system, the transient phase of the analog processing is likely to have little effect on the closed-loop control. It is then conceivable to operate the neurochip continuously in closed-loop, with fully analog processing from the sensors to the actuators. By reducing the time-delay encountered by the signal from the sensors to the actuators, an all analog closed-loop processing would significantly reduce the phase stability margin needed by a practical implementation of the neurocontroller to achieve the desired control objectives, and would therefore increase the performance of the neurocontrol.

For a crude estimate of the sensitivity of the neurocontroller performance on the accuracy of the output signal, the neuroprocessing was still simulated with the activation function (10), yet by allowing random fluctuations of the output signal, o , around its expected value, o_t . The input and output data were held in input and output buffers during 5ms periods, and the output data were randomly selected between $o_t - 1/2^{p-2}$ and $o_t + 1/2^{p-2}$ for a p -bit accuracy of the output signal. Closed-loop responses were simulated with an additional 40ms time-delay in both control channels, and for various output signal accuracies. With these setting conditions, the neurocontroller performed satisfactorily both in terms of command tracking and control requirement, for an accuracy of 8 bits or more in the output signal.

Fault-Tolerance. To analyze the tolerance of such analog VLSI neurocontrollers to degradations or failures of their electronic components, the analog processing was simulated with the neuron transfer function (10) for a gain of 1, and with an 8-bit weight accuracy.

Weight Degradations/Failures. The values of the stored weights will change in time because of the dipoles resulting from charge movement between the control gate and the floating gate of the EEPROM. These dipoles compensate for the electric field that was created by the EEPROM programming, which offsets the voltages of both the *weight cell* and the *reference cell*. Over several years, charge movements would ultimately cancel each cell voltage, and all differential weight voltages would go to zero. Because this electrical reaction is more intense as the cell voltages are high, weights of large magnitude degrade faster than weights of small magnitude. To be on the conservative side however, closed-loop neurocontrol was simulated for a given weight accuracy irrespective of the relative magnitudes of the weights. For a degradation leading to a p -bit weight accuracy after time τ , the weight values w_τ are therefore approximated as:

$$w_\tau = (w_0 - \frac{\text{sgn}(w_0)}{2^{p-1}}) \theta(|w_0| - \frac{1}{2^{p-1}}), \quad (11)$$

w_0 being the initial values set stochastically around the targets w_i with an 8-bit accuracy, and $\theta(x)$ being the step function $\theta(x) = 1$ if $x > 0$, and $\theta(x) = 0$ otherwise. The simulation results are illustrated in the closed-loop responses of Fig.7 for $p = 6, 7, 8$ and 9 , indicating small deviations from the ideal closed-loop responses for an accuracy of 8 bits or more in the *collective* degradation of the weights.

Anomalous charge leakages may also occur in certain floating gates of the synapse multiplier circuits as a result of hardware defects, and cancel the voltages of the defective EEPROM cells. Charge leakages in *both* the weight cell and the reference cell of the same synapse multiplier circuit would thus cancel the differential weight voltage (or weight) of the synapse. In the case where only one of the two EEPROM cells of a synapse multiplier circuit would be defective, the differential weight voltage would be that of the non-defective cell and could possibly take any value between -2.5 volts and 2.5 volts (corresponding to extremal *weight* values of -1 and +1).

The effect of such weight degradations or failures on the neurocontroller performance was estimated through Monte Carlo simulations of closed-loop responses by setting the value of a single arbitrary weight to -1, 0, or +1. The simulation results indicated a much lower fault-tolerance of the neurocontroller when weights degrade to -1 and +1 than when they degrade to 0. A remedy may be given by innovating training algorithms that would maximize the degree of distribution of the neural processing throughout the entire set of weights and neurons. Another alternative that is more in line with the scope of this work is to investigate the possibility of maintaining neurocontrol performance in the presence of weight degradations/failures by retraining the defective chip to achieve the desired control objectives. In view of the significant number of training parameters and the empirical nature of the training process, it is imperative that the retraining procedure be *automated* by *pre-programming* the choice and sequence of the training parameters, if it is to truly provide *fault-tolerance* to an *autonomous* neurocontrol architecture.

In the proposed automated retraining procedure, the commanded trajectories to be used for retraining are randomly generated following the method described in Section 2. The initial value of the steepest-descent parameter is fixed and very small. The weights are updated only after backpropagating the integral of the objective function over an entire commanded trajectory [6-7]. As learning takes place, the value of the steepest-descent parameter is progressively increased in a *programmed* manner. This automated retraining procedure was subsequently tested over each of the aforementioned weight failures/degradations for the fifteen most critical weights identified in Section 3, and shown in Fig.2. The performance of the retrained neurocontroller was found to be excellent in all 45 cases of degradations/failures except one for which the closed-loop responses presented an offset in steady-state tracking.

The proposed automated retraining procedure was also tested over the aforementioned types of degradations/failures of the thresholds of the eight most critical neurons identified in Section 3, and shown in Fig.2. The performance of the retrained neurocontroller was found to be excellent in all 24 cases of threshold degradations/failures.

Neuron Failures. Since the neurons are implemented as high-gain operational amplifiers with a sigmoidal-type of activation function, failures that are most likely to occur correspond to output values of -1, 0, or +1. The *same* automated retraining procedure was tested over these failures for the eight most critical neurons identified in Section 3, and shown in Fig.2. The performance of the retrained neurocontroller was found to

be excellent in all 24 cases of neuron failures, as illustrated by the pitch-rate response in Fig.8 when the output of neuron $n_{3,6}$ is forced to +1 during the entire closed-loop operation. The pitch-rate response of the non-retrained defective neurocontroller is also shown in Fig.8 for comparison with that of the retrained neurocontroller. The performance increase following retraining indicates that the backpropagation algorithm rearranges the weights to counter the effect of the defective neuron. The weights of the output layer before and after retraining are shown in Fig.9. As expected, the strength of the interconnection $w_{4,2,3,6}$ which connects the output of the defective neuron to the δTV -neuron is significantly attenuated after retraining. The other weight changes are distributed among the weights connecting into the δTV -neuron, the largest increase being for the threshold of the δTV -neuron. This indicates that, during the retraining, the backpropagation algorithm also utilizes the threshold parameters to achieve the desired control objectives in the presence of a neuron failure. The fact that the weights connecting into the WF -neuron vary very little in Fig.9 comes with little surprise in view of the influence diagram of Fig.2.

It is noted that the loss of control stability due to chip failure can be very severe, as in the example of Fig.8, and that real-time analysis of the actual closed-loop response of the system may not allow an early detection of such chip failures. Alternate fault-detections could then be provided by monitoring changes of individual weight values and individual neuron activities. A fault-detection scheme taking advantage of the very fast forward-processing of the chip (in the order of μs) would consist of continuously analyzing *simulated* closed-loop system responses to characteristic pilot input commands, e.g. the demanding pilot input command chosen to illustrate the neurocontroller performance in Figs.5-8.

Finally, due to the global learning nature of backpropagation networks, retraining the defective neurocontroller may require to estimate its performance over a broad range of command inputs that extends beyond those actually being requested by the pilot in closed-loop control. Whether the control input to the actuators will have to be provided by an auxiliary neurocontroller while the defective neurocontroller is being retrained, or whether the control input can still be generated by the relearning defective neurocontroller is therefore an important question that needs to be addressed in future works.

5. Conclusion.

The performance and fault-tolerance of analog parallel distributed implementations of a multivariable aircraft neurocontroller were analyzed by using a simplified scheme of analog processing based on the functional architecture of the ETANN chip. The neurocontroller was found to be only *partially* fault-tolerant to weight or neuron failures. Although some fault-tolerance could be gained by using the *Locally Distributed Weight and Neuron Approaches*, reaching a satisfactory level of fault-tolerance required retraining the defective neurocontroller by an *automated* procedure based on a pre-programmed choice and sequence of the training parameters.

As analog VLSI matures, greater learning speeds and therefore faster retraining can be expected from analog neural networks. The objective function which contains the control objectives could be minimized with a faster convergence algorithm to reduce the retraining time. Addressing concurrently these hardware and software issues would help decide the feasibility of acquiring fault-tolerance in real-time, i.e. through an *automated retraining procedure within the control loop*.

Acknowledgments.

The authors thank Mr. Hernan Castro for very informative/stimulating discussions on the ETANN chip.

References.

- [1] J. Alspector, et al., *A Neuromorphic VLSI Learning System*, 1987 Stanford Conference, Advanced Research in VLSI, pp.313-349, 1987.
- [2] M. Holler, S. Tam, H. Castro, and R. Benson, *An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses*, Int. Joint Conf. on Neural Networks, Washington D.C., June 1989.
- [3] Intel Corporation, *80170NX Electrically Trainable Analog Neural Network*, Data Sheet, June 1991.
- [4] T. Troudet, S. Garg, and W. Merrill, *Design and Evaluation of a Robust Dynamic Neurocontroller for a Multivariable Aircraft Control Problem*, Int. Joint Conf. on Neural Networks, Baltimore, MD, June 1992.
- [5] S. Garg, D. L. Mattern, and R. E. Bullard, *Integrated Flight/Propulsion Control System Design Based on a Centralized Approach*, Journal of Guidance, Control and Dynamics, Vol.14, No.1, Jan.-Feb. 1991.
- [6] T. Troudet, S. Garg, and W. Merrill, *Neural Network Application to Aircraft Control Design*, AIAA Guidance, Navigation and Control Conference, New Orleans, August 1991.
- [7] T. Troudet, S. Garg, D. Mattern, and W. Merrill, *Towards Practical Control Design Using Neural Computation*, Int. Joint Conf. on Neural Networks, Seattle, WA, July 1991.

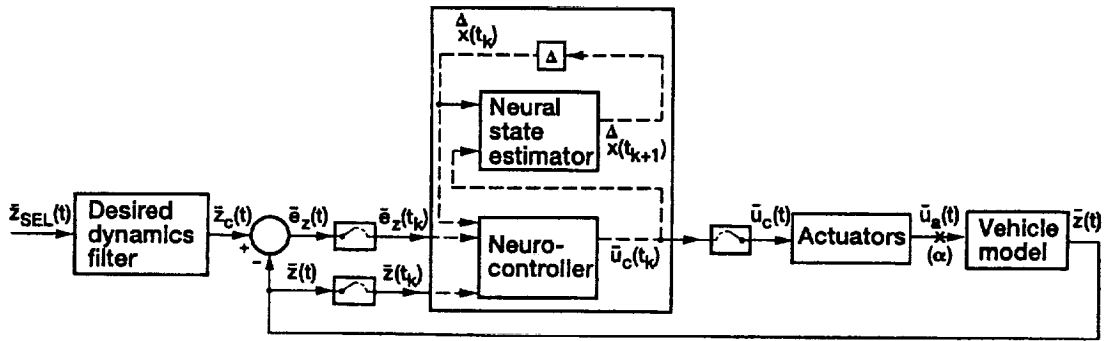


Figure 1.—Closed-loop evaluation architecture of the dynamic neurocontroller.

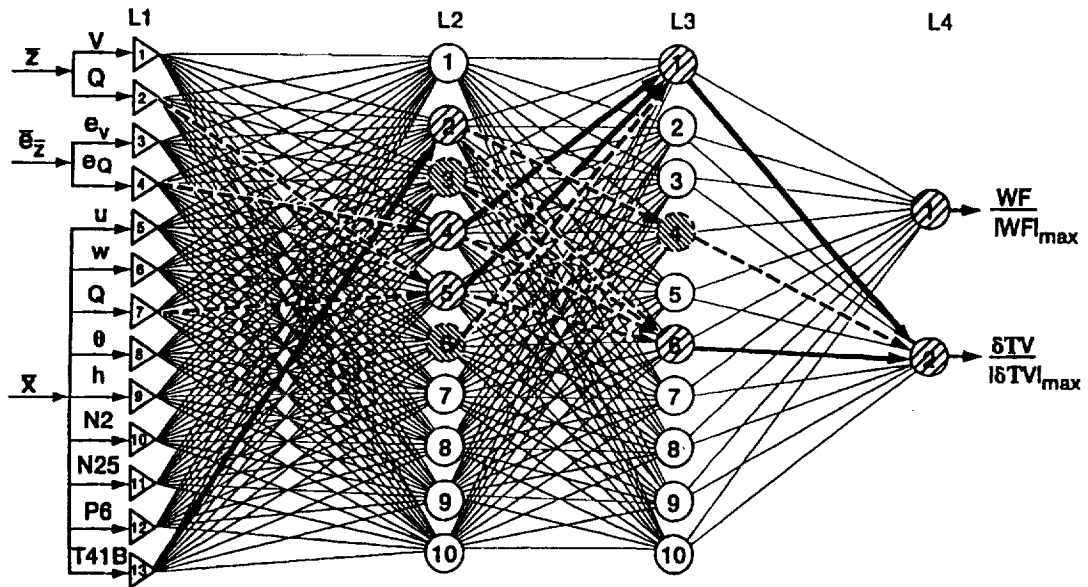


Figure 2.—Contribution of individual weights and neurons to the neural computation.

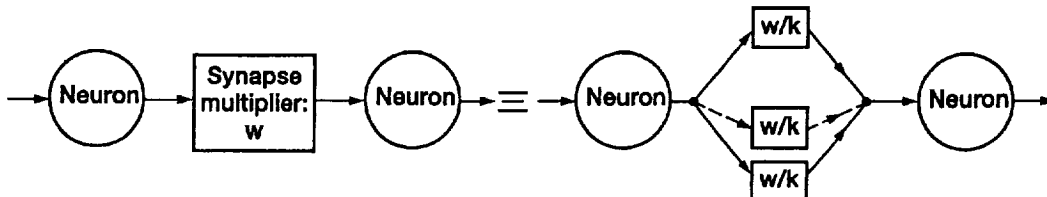


Figure 3.—LDWA^(k): Locally distributed weight approach.

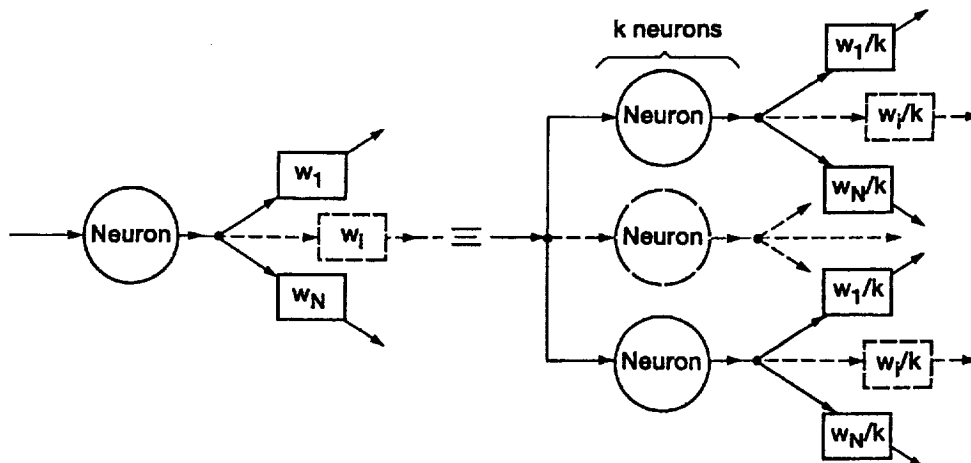


Figure 4.—LDNA^(k): Locally distributed neuron approach.

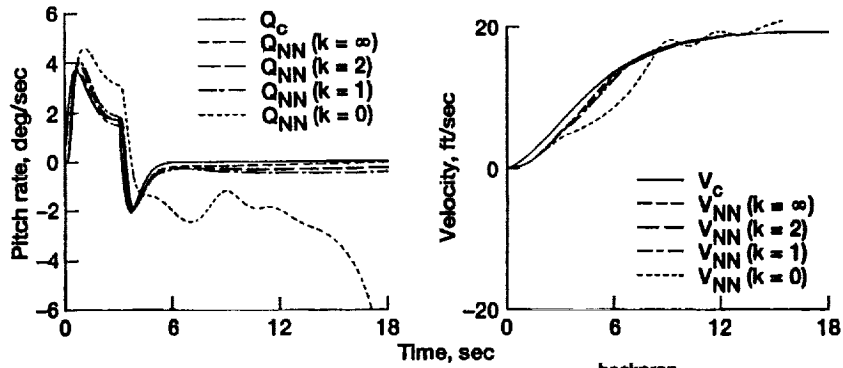


Figure 5.—Closed-loop responses with $w_{4,2;3,1}^{backprop} / w_{4,2;3,1} = k/(k+1)$ for $k = 0, 1, 2, \text{ and } \infty$.

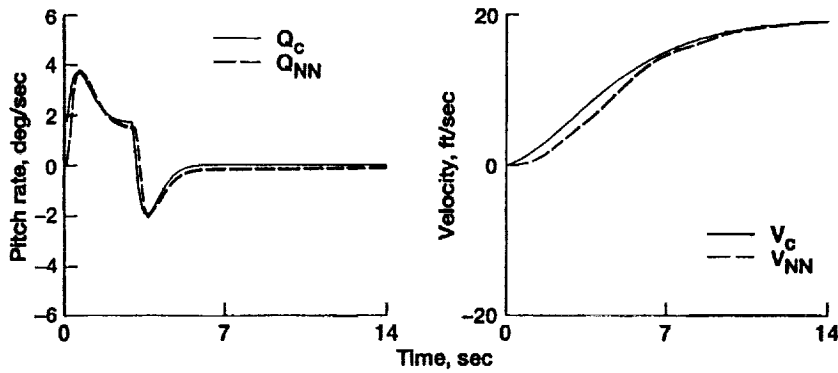


Figure 6.—Closed-loop responses with an 8-bit accuracy in the setting of the weights (20 random settings).

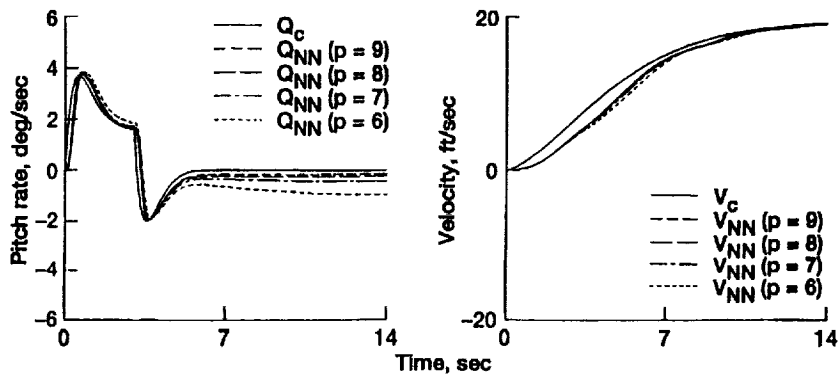


Figure 7.—Closed-loop responses with a p-bit accuracy in the collective degradation of the weights due to time-decay.

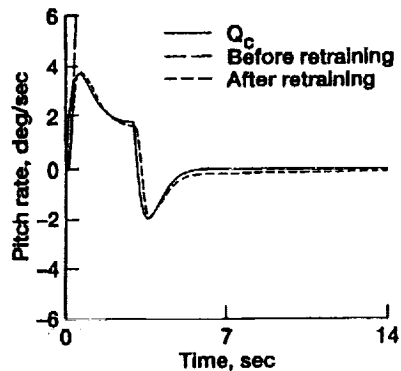


Figure 8.—Pitch-rate response for $O_{36} = 1$ before/after retraining.

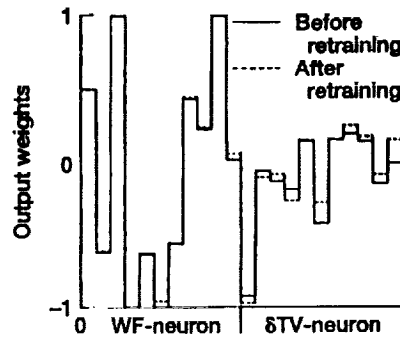


Figure 9.—Weights of output layer before/after retraining.



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1992	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Analysis of Fault-Tolerant Neurocontrol Architectures		5. FUNDING NUMBERS WU-505-62-50	
6. AUTHOR(S) T. Troudet and W. Merrill			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		8. PERFORMING ORGANIZATION REPORT NUMBER E-7210	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-105898	
11. SUPPLEMENTARY NOTES Prepared for the 31st Conference on Decision and Control sponsored by the Institute of Electrical and Electronics Engineers, Tucson, Arizona, December 16-18, 1992. T. Troudet, Sverdrup Technology, Inc., Lewis Research Center Group, 2001 Aerospace Parkway, Brook Park, Ohio 44142. W. Merrill, Lewis Research Center. Responsible person, T. Troudet, (216) 433-8524.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 08		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The fault-tolerance of analog parallel distributed implementations of a multivariable aircraft neurocontroller is analyzed by simulating weight and neuron failures in a simplified scheme of analog processing based on the functional architecture of the ETANN chip. The neural information processing is found to be <i>only partially</i> distributed throughout the set of weights of the neurocontroller synthesized with the backpropagation algorithm. Although the degree of distribution of the neural processing, and consequently the fault-tolerance of the neurocontroller, could be enhanced using <i>Locally Distributed Weight and Neuron Approaches</i> , a satisfactory level of fault-tolerance could only be obtained by retraining the degraded VLSI neurocontroller. The possibility of maintaining neurocontrol performance and stability in the presence of single weight or neuron failures was demonstrated through an <i>automated retraining procedure</i> of the neurocontroller based on a pre-programmed choice and sequence of the training parameters.			
14. SUBJECT TERMS Fault-tolerance; Neural networks		15. NUMBER OF PAGES 12	
		16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT