Reliable, Trainable Networks for Computing and Control

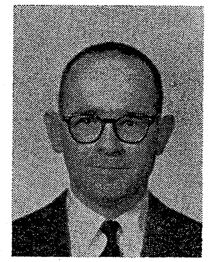
Bernard Widrow • James B. Angell

Stanford University

ELECTRONIC NETWORKS and systems which can perform their intended functions despite defective components, subassemblies, or interconnections within them provide an interesting possibility for enhancing system reliability. In the various sections of this paper, a number of proposed techniques for achieving this goal will be reviewed briefly, after which more detailed consideration will be given to networks of adaptive, or "trainable," linear elements called *Adalines.* Such networks have exhibited great tolerance to internal imperfections, and possess many desirable properties as data-processors which can improve with experience.

The advent and growth of microsystem electronics, for which decreased size and increased density of components are major goals, have added both encouragement and impetus to the search for system organizations which are tolerant of defects. The encouragement comes from the potentially small space, weight, and cost involved in providing the additional, redundant components which these reliable system organizations require. The impetus is provided by the difficulty in replacing defective parts and by the problems of initial yield in a microsystem in which a large number of components, integrally mounted together, have probably been fabricated en masse. Hence, it is contended that interest in microelectronics is highly compatible with the system studies described here.

Certainly any system that can function despite internal imperfections will have more parts than a minimal system since the job of the imperfect parts must be borne by other, redundant parts which are still functioning. Some techniques of exploiting redundant parts are far more efficient than others, when measured by the amount of redundancy which must be added to provide a given degree of immunity to overall failure. Redundant systems might be classified as *active* or *passive*;[1] the first term applies to systems in which reserve parts are switched in to replace defective ones (the spare tire is a nonautomatic example), whereas in passive redundancy the parts are so connected that, when one part fails, its function is automatically assumed by other parts



Dr. Widrow is Associate Professor of Electrical Engineering currently engaged in research and teaching in systems theory, control theory, adaptive logic, and adaptive control systems. He is, in addition, President of Memistor Corporation, manufacturer of memistors and adaptive computing machines. A 1951 graduate of Massachusetts Institute of Technology, he received his Master's and Doctor's degrees from M.I.T. in 1953 and 1956. From 1951 to 1956 he was on the staff of M.I.T.'s Lincoln Laboratory and a Research Assistant in the Department of Electrical Engineering. For the next three years he taught radar, theory of sampled data, systems, and control theory at M.I.T. He accepted his present position at Stanford in 1959. Dr. Widrow is a member of AIEE, IRE, and Sigma Xi.

Dr. Angell is Professor of Electrical Engineering with his principal research interest in components for adaptive systems. He received his B.S., M.S., and Sc.D. degrees from Massachusetts Institute of Technology, the first two in 1946, the last in 1952. From 1946 to 1951 he was a Research Assistant at M.I.T.'s Research Laboratory of Electronics, studying noise in tracking radars. In 1951 he joined Philco Corporation's Research Division as Manager of solid-state circuit research, concerned with the evaluation and application of solid-state devices in high-speed and high-frequency service. Dr. Angell joined the Stanford faculty in September 1960. He is a member of IRE, AIEE, and several honorary societies.

The work reported here was supported by the Aeronautical Systems Div., Air Force Systems Command, under Air Force Contract No. AF 33(616)-7726, and jointly by the U.S. Army Signal Corps, the U.S. Air Force, and the U.S. Navy (ONR), under Contract No. Nonr 225(24).

"A new type of logic, adaptive logic, is being devised which promises to play a significant role in the future development of computers. Not designed in detail in the usual way, it can, instead, learn to function by being trained by the designer, or it can spontaneously learn from its environment. In a sense, such systems are inherently reliable. They can adapt to their own internal failures. Systems containing adaptive vote-takers are bridges between conventional fixed-logic systems and systems adaptive 'from the ground up.'"

(dual tires on the rear end of a bus, for example).

In the next two sections of this paper, we shall compare certain forms of passive redundancy, and consider one, based on an adaptive vote-taker, which might be considered semiactive, in that defective, or failed, elements are gradually deleted from a system that was originally purely passive.

The final three sections of the paper are concerned with the properties and realization of adaptive networks comprising interconnected *ada*ptive, *line*ar threshold elements called *Adalines.* An Adaline has a number of binary inputs (1 or -1) and a single binary output that is positive when, and only when, the weighted sum of the binary inputs exceeds a threshold level. The weighing factors and the threshold level are changeable analog quantities, thereby permitting adaptation to take place. These networks have many desirable properties as statistical, trainable pattern classifiers, including the ability to provide a best binary "decision," based on past training experience, to a previously unseen problem. Various types of generalization, such as trained-in insensitivity to noise in patterns and to pattern rotation, size, and displacement, are discussed in Section 3. The use of a single Adaline as a trainable pattern classifier in a control system is considered in Section 4. In Section 5, two components that provide the required variable weight with permanent memory are described, and the application of one of them to an adaptive element is outlined.

(1) Reliability Via Redundancy

Various schemes which have been proposed for improving the reliability and life of data-processors, by means of redundant information or structure, are:

(1) Error-detecting codes.

(2) Comparison of parallel or sequentially duplicated outputs.

(3) Stand-by duplicate equipment.

(4) Error-correcting codes.

(5) Redundant switching networks.

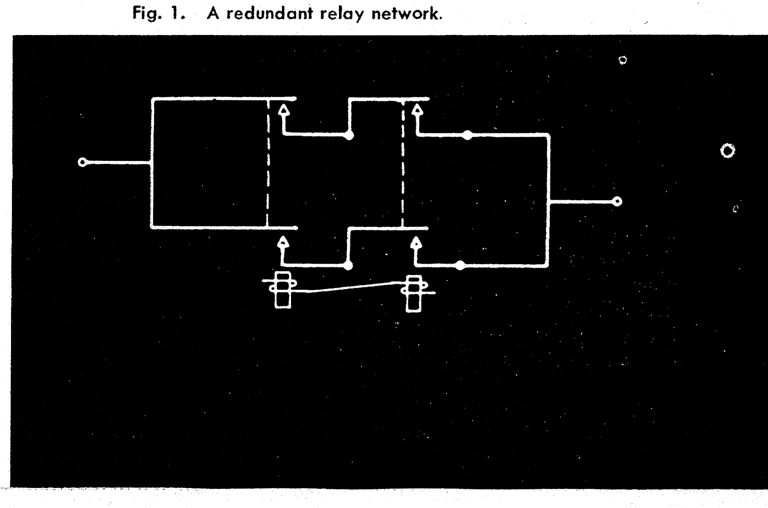
(6) Majority vote among an odd number of parallel outputs.

(7) Adaptive majority vote among parallel outputs.

The first two items of this list do not of themselves provide for reliability, but merely indicate when trouble has arisen; after such an indication has been given, additional steps, such as switching to stand-by duplicate equipment or repairing the source of the trouble, must be taken. On the other hand, the last four items do permit continuous operation, in that defective parts within a system do not hamper operation (unless there are too many of them). Not included in this list is the well-known marginal checking routine for computers, in which a known, diagnostic program is processed while the computer is stressed (usually by reducing the supply voltage to the machine). This technique does not involve

(*Continued on page 116*)

Fig. 1. A redundant relay network.



(Continued from page 79)

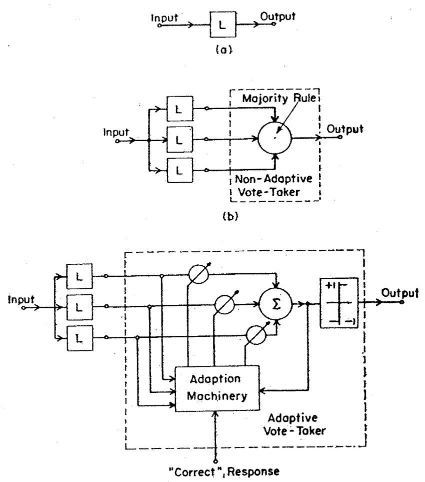


Fig. 2. Fixed-majority and adaptive vote-takers.

redundancy, but it does take time.

Additional discussion is given in the remainder of this section on error-detecting and error-correcting codes, and on the Shannon-Moore redundant relay networks, in order to exemplify the ideas involved here. The use of fixed majority voters, as proposed by von Neumann,[4] and of adaptive majority voters, as described by Pierce,[1] is considered in Section 2.

Error-detecting codes in digital computers require that additional binary bits be added to each word, or block of information, that is stored or processed in such a way that errors (usually a change in one or more bits within a word) give rise to patterns which are unacceptable to the computer. The parity check is the simplest and most widely used of the single-error detecting codes. In a parity check, a single bit (1 or 0), usually at the beginning or end of a word, is set so that the total number of 1's in the word, including the parity bit, is an even number. The computer is then organized to recognize only words with an even number of 1's; if it encounters a word with an odd number of 1's, it stops and indicates that an error has been made. (Obviously, the role of the even and odd numbers could be interchanged.) A single error (the loss or addition of a 1) within a word will be detected by the computer, although any even number of such errors would go unnoticed.

Error-correcting codes represent an extension of the idea introduced in the preceding paragraph. One way of viewing the parity check is to realize that every acceptable word differs from every other by changes in at least two bits. If additional bits are used, it is possible to arrange them so that all acceptable words are mutually different by at least three bits. A single error then causes no confusion because it is certain that the desired word can be found by temporarily changing the bits, one at a time, in the unacceptable word until an acceptable word is found. Thus, we have single-error correction. Indeed, two errors can be thus detected, so that by a relatively minor addition of information one provides for single-error correction and double-error detection. Through efficient use of the additional bits in an error-correcting code, it is possible to have relatively simple decoding that tells immediately (rather than by sequential trial and error) if a word is correct, and, if not, which bit is in error.[2] Table I shows the number of additional bits required for various (noncorrected) word lengths in order to provide for this single-error correction.

Redundant switching networks were introduced by Shannon and Moore[3] as a means of assembling reliable switching networks out of relatively less reliable relay contacts. Their ideas are illustrated here in the simple relay-contact network shown in Fig. 1. By connecting contact pairs, all nominally opened or closed together, in series, the desired state is ensured despite the fact that one pair of contacts may weld together permanently. Similarly, by connecting pairs, or series chains, in parallel, we ensure operation despite the permanent opening (via dust or corrosion) of one pair of contacts. The idea can be extended to larger series—parallel, or lattice connection of redundant contact pairs, which can be optimized by statistical knowledge of the manner in which failures occur. Although the Moore-Shannon proposal was phrased in terms of relay contacts, it is apparent that the same ideas can be applied to networks of other switching devices, such as diodes and transistors. However, for transistor switching, the required input-drive power and additional cost of this form of redundancy at the component level have, thus far, precluded its extensive use.

Principles and comparisons of *majority vote* and *adaptive majority vote* among parallel redundant outputs are summarized in Section 2. The adaptive networks considered in the remainder of the paper have demonstrated a remarkable tolerance to defective internal components (with somewhat reduced performance capability), even though they are not organized with this property as the most important consideration.

Table 1. Information Redundancy Needed for Single-Error Correction Code

Word Length (Binary Bits)	Additional Bits
1	2
2 to 4	3
5 to 11	4
12 to 26	5
27 to 57	6

(2) Majority Vote and Adaptive Majority Vote

The reliability of digital systems can be made arbitrarily high by the use of redundant circuits. A combination of adaptive circuits with redundant circuits allows the same high reliability with a drastic reduction in the amount of redundancy. The vote-taking scheme of von Neumann[4] can be modified by the use of an *adaptive vote-taker* which weights more heavily the votes of the subsystems which it learns are the more reliable.

A single logical operator is shown in Fig. 2a. Any imperfection in this logic block L will cause error in its output. In the system of Fig. 2b, three logic blocks and a majority-rule element are utilized to perform the function of the simple logic block L of Fig. 2a; here, however, an error in one logic block out of the three will not cause an error in the final output. If errors in the blocks L are random and unrelated to each other, the majority-rule vote-taker is the best choice. Should one of the logic blocks become defective, equal vote weight would no longer remain the best policy. In Fig. 2c, vote weights are adjusted by an adaptive process to minimize error probability. The vote weight for a defective logical element is automatically brought to zero, and the defective element is thus deleted from the system. The nature of the adaptive process will be explained later in thi paper.

In Fig. 3a, a series of N functionally independent logic blocks is shown. Error in any one of these blocks will, in general, cause error in the output signal, whereas only error in an entire section will cause output error in the system of Fig. 3b. The series of vote-takers that couple each section to the next has been called a *restoring organ* by von Neumann. The signals on the output lines of the restoring organs are generally more reliable than those on the input lines. Notice that the system of Fig. 3b tolerates errors not only in the L-blocks, but also in the vote-takers, which may be fixed majority-rule types or adaptive types.

The reliability of various system configurations containing adaptive vote-takers has been analyzed extensively by Pierce.[5] The effects of redundancy and adaptation upon error rate and lifetime extension are suggested in the following examples.

Let the systems of Fig. 3 have 32 interdependent logic sections, and let the redundancy factor of Fig. 3b be three times, as shown. Assume that the vote-takers themselves are perfect (their unreliabilities could be lumped into that of their succeeding logic blocks). Table 2 is a comparison of error rates. Let each logic block make one error in 10^9 calculations.

Assume that the probability of component survival drops exponentially with time (Table 3). Let there be 32 interdependent logic sections, and let the redundancy factor be 3. The mean lifetime of a single logic block is normalized to 1.

With increased redundancy and with adaptation, it is possible to construct a large complex system having a greater mean lifetime than the mean-lifetime of its component parts.

(3) Adaptive Pattern-Recognizing and Logic Systems

It was seen that adaptive decision elements, also called vote-takers, are like automatic repairmen constantly on duty in their respective locales, always ready to delete parts that become defective. This type of self-repair makes possible optimal use of the remaining functioning components, and is especially applicable to systems of fixed logical structure. A new type of logic, adaptive logic, is being devised which promises to play a significant role in the future development of computers. Not designed in detail in the usual way, it can, instead, learn to function by being trained by the designer, or it can spontaneously learn from its environment. In a sense, such systems are inherently reliable. They can adapt to their own internal failures. Systems containing adaptive vote-takers are bridges between conventional fixed-logic systems and systems adaptive "from the ground up."

A self-contained, automatically adapted logical element called the Adaline *neuron* has been developed for pattern-recognition systems and as a basic element for adaptive logical circuits.[6] This element would serve directly as an adaptive vote-taker. (Such an application is discussed in detail below.) A block diagram of Adaline is shown in Fig. 4. (Note the similarity to the decision element of Fig. 2c.) It represents a flexible threshold-logic circuit having input lines, a single output line, and an input line, called the *desired output*, which is actuated during training only.

The binary input signals to Adaline have values of $+1$ or -1, rather than the usual values of 1 or 0. Within the neuron, a linear combination is formed of the input signals, each of which is multiplied by a certain weighting factor. The weights are the gains $a_1, a_2, \ldots a_n$, which can have both positive and negative values. The output signal is $+1$ if the weighted sum is greater than a certain threshold, and -1 otherwise. The threshold level is determined by the setting of a_0, whose input is permanently connected to a $+1$ source. Varying a_0 varies a constant added to the linear combination of input signals.

For fixed-gain settings, each of the 2^n possible input combinations could cause either a $+1$ or a -1 output. Thus, all possible inputs are classified into two categories. The input-output relationship is determined by choice of the gains $a_0, a_1, \ldots a_n$. In the adaptive neuron, these gains are set during the training procedure.

In general, there are 2^{2^n} different input-output relationships, or truth functions, by which the n binary input variables can be mapped into a single binary output variable. Only a subset of these relationships, the linearly separated truth functions, can be realized[7] by a single neuron of the form shown in Fig. 4. Although this realizable subset is not all-inclusive, it is a very useful subset, and it is "searchable," in that optimum gain settings for a given truth function can usually be found by a convergent iterative process.

Application of this neuron in adaptive pattern classifiers was first made by Mattson.[8] He has shown that complete generality in choice of switching function could be achieved by combining these neurons. He devised an iterative digital computer routine for finding the best set of a's for the classification of noisy geometric patterns. An iterative procedure having similar objectives has been devised by Widrow and Hoff. This procedure is simple to implement, and can be analyzed by statistical methods that have been developed for the analysis of adaptive sampled-data system.[9]

An Adaptive Pattern Classifier

An adaptive pattern-classification machine (Fig. 5) has been constructed for the purpose of studying and illustrating adaptive behavior and artificial learning. It represents a single manually adapted Adaline neuron.

During a training phase, simple geometric patterns are fed to the machine by setting the toggle switches in the 4 × 4 input switch array. All gains, including the threshold level, are to be changed by the same absolute magnitude so that the analog error (the difference between the desired meter reading and the actual meter reading) is brought to zero. This is accomplished by changing each gain in the direction that will diminish the error by 1/17. The 17 gains may be changed in any sequence, and, after all changes are made, the error for the present input pattern is zero. The weights associated with switches up ($+1$ input signals) are incremented by rotation in the same direction as the desired meter needle rotation; the weights connected to switches in the down position are incremented opposite to the desired direction of rotation of the meter needle. The next pattern and its desired output are then presented, and the error is read. The same adjustment routine is followed and the error brought to zero. If the first pattern were reapplied at this point, the error would be small but not necessarily zero. More patterns are inserted in like manner. Convergence is indicated by small errors (before adaptation), with small fluctuations about stable weights. Note that adaptation is indicated even if the quantized neuron output is correct. If, for example, the desired re-

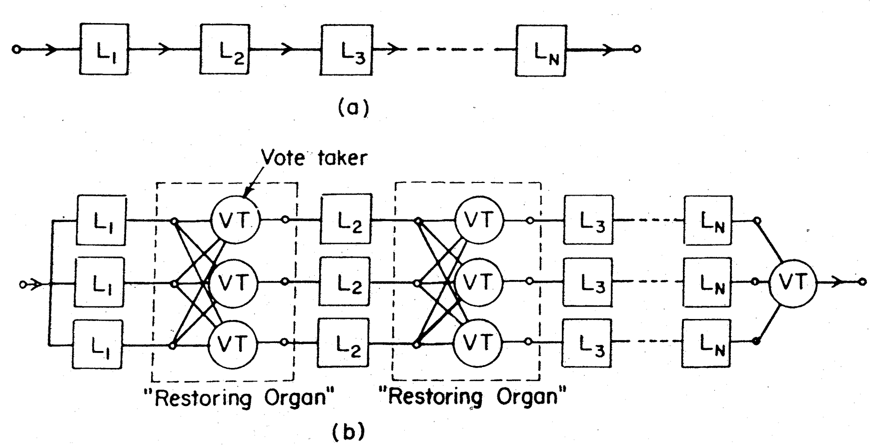


Fig. 3. Redundant network with vote-taker.

Table 2

Nonredundant system (Fig. 3a)	Redundant system, fixed majority-rule vote-takers (Fig. 3b)	Redundant system, adaptive vote-takers (Fig. 3b)
32 errors per 10^9 calculations	11 errors per 10^9 calculations	2 errors per 10^9 calculations

Table 3

Nonredundant system (Fig. 3a)	Redundant system, fixed majority-rule vote-takers (Fig. 3b)	Redundant system, adaptive vote-takers (Fig. 3b)
Lifetime $= \frac{1}{32}$	Lifetime $= \frac{11}{32}$	Lifetime $= \frac{16}{32}$

sponse is $+1$, the neuron is adapted to bring the analog response closer to the desired response, even if the analog response is more positive than $+1$.

The results of a typical adaptation on six noiseless patterns is given in Fig. 6. During adaptation, the patterns were selected in a random sequence and classified into three categories. Each T was to be mapped to $+30$ on the meter dial, each G to 0, and each F to -30. As a measure of performance, after each adaptation, all six patterns were read in (without adaptation) and six errors were read. The sum of their squares denoted by $\Sigma\epsilon^2$ was computed and plotted. Fig. 6 shows the learning curve for the case in which all gains were initially zero. The theoretical time constant is 17 patterns, equal to the number of weights adapted.[6]

How many patterns or stimuli can the single adaptive neuron be trained to react to correctly at a time? This is a statistical question. If, within a large group of patterns, those which are to give the $+$ response are similar to each other and dissimilar to those which are to give the $-$ response, the neuron has little trouble in adapting to make the desired distinctions. If, however, two patterns that differ by one bit are to give opposite responses, the critical weight must have a large value and be of appropriate sign. If two other similar patterns are to be inserted to give opposite responses, and the same weight is the critical one—but here the necessary sign requirement for this weight is opposite to the previous requirement—clearly the set of only four patterns will not be linearly separable. A series of experiments was devised by Koford where patterns containing unbiased random bits and random desired responses were applied to Adalines with varying number of inputs. *It was found that the average number of random patterns that can be absorbed by an Adaline is equal to twice the number of weights.* This is one basic measure of memory capacity.

Madaline, a Parallel Network of Adalines

Storage capacity in excess of that of a single Adaline can be readily achieved by use of parallel multineuron networks. Several neurons can be used to assist each other in solving problems by automatic load-sharing.

The configuration in Fig. 7 shows five Adalines having parallel-connected inputs. Their five outputs are connected to a majority-rule element whose output is the system output. One procedure for training this network is the following. A pattern is inserted, and if the response of the majority element is the desired response, no adaptation takes place. But if the desired response is $+1$ and three of the five Adalines read -1 for the given input pattern, one of the latter three must be adapted to the $+1$ state. The one that is adapted is the one whose confidence level (the sum before quantization) is closest to zero—i.e., the one whose analog response is closest to the desired response. If more of the Adalines were originally in the -1 state, enough of them would be adapted to the $+1$ state to make the majority decision $+1$. The ones adapted would have had confidence levels closest to zero. This adaptation procedure is symmetric with respect to adaptation when the desired response is -1. Differences in initial conditions and the results of subsequent adaptation cause the various neurons to take "responsibility" for certain parts of the training problem. The basic principle of load-sharing is summarized thus: *Assign responsibility to the neuron or neurons that can most easily assume it.*

In Fig. 7, the "job assigner," a purely mechanical process, assigns responsibility during the training process by transferring the desired-response adapt commands to the selected Adalines. The job assigner utilizes confidence level information.

The adaptive system of Fig. 7 was suggested by common sense, was tested by simulation, and was found to work very elegantly. It was subsequently proved by Ridgway[10] in his doctoral thesis that if a set of weights exists that will solve the training problem, then this system will converge on a solution. The essence of the proof lies in showing that the probability of a given neuron's taking responsibility for adaptation to a given input stimulus-desired response is greatest if that neuron had taken such responsibility during the previous adapt cycle when the stimulus was most recently inserted. The division of responsibility stabilizes at the same time that the responses of the individual neurons stabilize to their share of the "load." When the training problem is not perfectly separable by this system, it can be shown that the adaptation process tends to minimize error probability.

In a sense, the Madaline (multiple Adalines) structure of Fig. 7 is two-layer—the first layer is of adaptive logic elements, the second of fixed logic. There are a variety of fixed-logic schemes that could be used on the second layer. Convergent adaptation procedures have

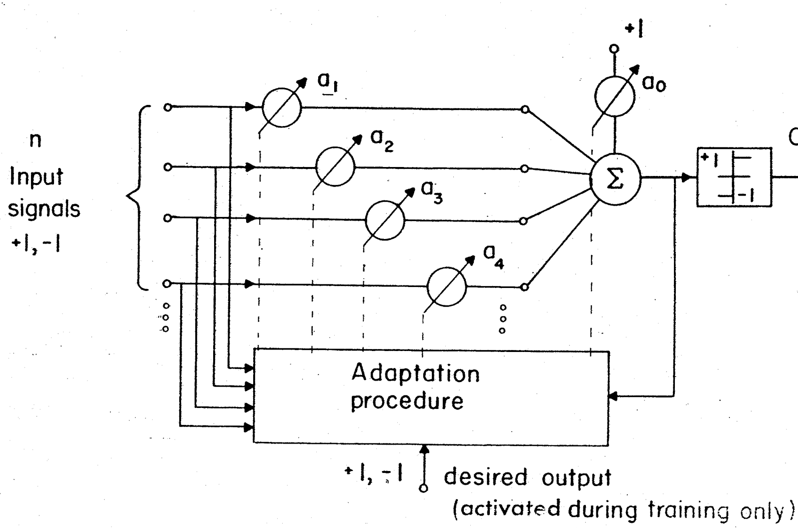


Fig. 4. An automatically adapted threshold element.

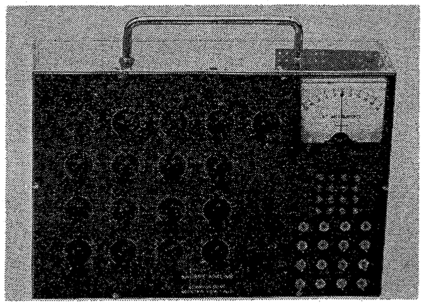


Fig. 5. An elementary learning machine.

been devised by M. E. Hoff, Jr. (to be described in his doctoral thesis) which can be used with all possible fixed-logic second layers. A simple fixed-logic element is an *OR* element. If any of the Adalines produce the $+1$ output, the *OR* element gives a system output of $+1$. During training, if the desired output for a given input pattern is $+1$, only the one neuron whose confidence level is closest to zero need be adapted if any adaptation is to be done—i.e., if all neurons give -1 outputs. If the desired output is -1, all neurons must give -1 outputs, and any giving $+1$ outputs must be adapted. Ridgway has also proved that this system is convergent.

The memory capacities of Madaline structures utilizing both the majority element and the *OR* element have been measured by Koford. Although the logic functions that can be realized with these output elements are different, both types of element yield structures with the same statistical storage capacity. The average number of patterns that can be adapted to by a Madaline equals the capacity per Adaline multiplied by the number of Adalines. The memory capacity is, therefore, equal to twice the number of weights.

Generalization Experiments With Adalines and Simple Networks of Adalines

With suitable pattern-response examples and the proper training procedures, generalizations can be trained into Adalines. The kinds of generalizations to be considered here are concerned with the training of Adalines to react to patterns and to be statistically insensitive to noise and rotation. Adalines can be forced to react consistently on a training set of patterns for all possible rotations, for example, and then they will react consistently to all rotations of new patterns never seen before and quite unrelated to the training set.

Generalization With Respect to Noise

In Fig. 8, a set of patterns is shown which was used in an experiment on generalization for insensitivity to noise. A single 3×3 Adaline was first trained on 100 noisy X, T, C, J patterns. This problem was solvable with a minimum error rate of 12 percent. The weights were returned to zero and ten patterns of the hundred were selected at random and trained into the Adaline. Then the response of the Adaline was tested on the full group of 100 patterns. On the average, a set of such experiments involving training with very small sample size produced an error rate of 23 percent. The theoretical error rate[6] for training with a number of patterns equal to the number of weights of the neuron is twice the minimum error rate or 24 percent, which checks nicely with the experimental result. If the number of training patterns was increased to 20, error rate would have been 18 percent. The number of patterns required to train an Adaline to discriminate noisy pattern equals several times the number of bits per pattern, roughly the statistical capacity of the neuron.

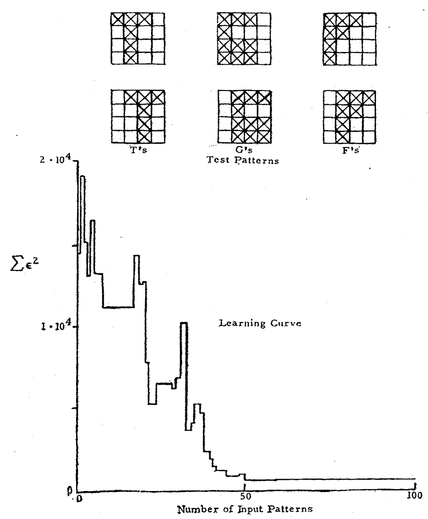


Fig. 6. Measurement of rate of adaptation.

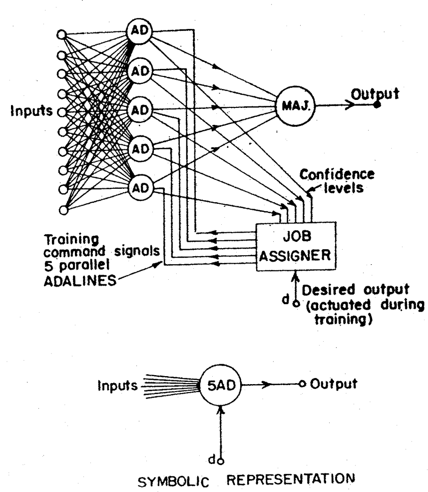


Fig. 7. Configuration of Madaline.

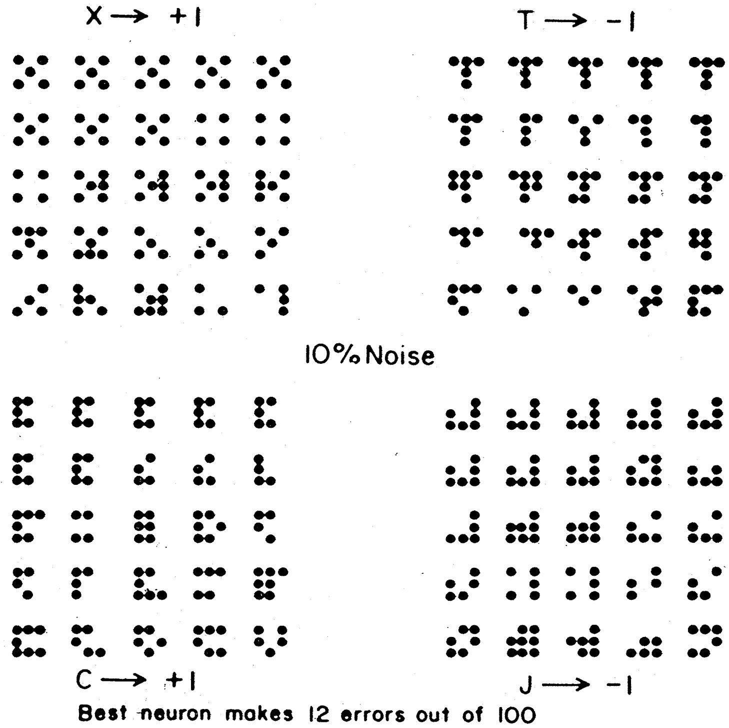
Generalization With Respect to Rotation of Patterns

Insensitivity to rotation by 90° is a characteristic that can be perfectly "trained-into" an Adaline. An experiment was devised (Fig. 9) using the 4×4 Adaline of Fig. 5. C's rotated in all four positions were trained-in to give the $+1$ response, while T's were trained-in to give the -1 response in all four rotations. The initial weights were set to zero and, during training, the minimum mean-square error adaptation procedure with an adaptive time constant of 32 patterns was utilized. The process converged with the desired responses trained-in precisely, and the set of weights shown in Fig. 9 resulted. Without further training, new patterns totally unrelated to the training patterns were inserted, and it was observed that not only were the decisions made by the Adaline perfectly consistent for each pattern over the four rotations, but that the four meter readings (confidence levels or analog outputs) for each pattern were identical. The reason for this is simple. Rotation of the weights by 90° yields an identical set of weights. Let the a-matrix represent the set of weights (excluding the threshold weight, which remains the same for all rotations). The superscript R represents rotation.

$$[a] = [a]^R = \left[[a]^R\right]^R = \left[\left[[a]^R\right]^R\right]^R \quad (1)$$

Other training patterns and other numbers of training patterns were used in this experiment and in each case, after convergence, the same symmetry expressed in Eq. (1) resulted automatically. Adaptation with a time constant long compared to the number of training patterns allows the neuron to retain responses to all the training patterns essentially equally. Minimization of mean-square error forces the response voltage to each training pattern in all voltage to be precisely $+1$ or -1. This forces the symmetry of Eq. (1).

How many specific responses on the average can be trained-in and yet have the neuron trained to be insensitive to 90° rotation for all patterns? The 4×4 neuron has a capacity of 32 patterns. Eight basic patterns on the average can be trained-in since each basic pattern must be inserted in all four rotations. Another point of view on this question was suggested by Hoff. The four encircled weights and the threshold shown in Fig. 9, once chosen, set the rest of the weights when the constraint of Eq. (1) is followed. There are four "degrees of freedom" plus the threshold freedom. The number of basic patterns that can be discriminated,



EXPERIMENT #	PATTERNS ADAPTED ON	NUMBER OF ERRORS	MISADJUSTMENT
1	95, 79, 07, 60 73, 61, 06, 02, 72, 26	25	$M = \frac{25-12}{12} = 108\%$
2	70, 69, 52, 55, 32 97, 30, 38, 87, 01	19	$M = \frac{19-12}{12} = 58\%$
3	65, 12, 84, 83, 34 38, 71, 66, 13, 80	20	$M = \frac{20-12}{12} = 67\%$
4	07, 42, 85, 88, 63 35, 37, 92, 79, 22	28	$M = \frac{28-12}{12} = 133\%$

Fig. 8. Training with noisy patterns.

therefore, corresponds to the capacity of a four-input neuron, which is eight patterns.

Other Generalization Experiments

In addition to the aforementioned training experiments for insensitivity to noise and rotation, a number of other tests have been made of the ability of Adalines to generalize on certain properties, including

(a) Insensitivity to vertical or horizontal translation.
(b) Insensitivity to pattern size.
(c) Direct sensitivity to rotation.
(d) Direct sensitivity to translation.

The last two experiments were attempts to teach the Adalines to distinguish (reverse the sign of the output) between 90° rotations or one-step translations of any input pattern. In general, the results are equivalent to those reported previously for generalization with respect to noise and rotation. In many of these experiments, the relationships among the weights that result are simple and could have been determined beforehand. However, it should be realized that, by using a single training routine, a wide variety of learning and generalization processes can be induced merely by designing appropriate sets of training patterns. All of these generalization experiments were done on the Adalines, such as that shown in Fig. 5, with initial weight settings of zero. In general, more training patterns would be required where it is impractical to set all weights to zero initially. The total number of patterns required per Adaline would be at least equal to the number of weights. The objective is to make the responses to patterns not specifically trained-in to depend only on the training experience and not on the initial conditions.

Adaline as an Adaptive Vote-Taker

Vote-taking is actually a form of pattern recognition. The array of output signals arising at each calculation cycle from a set of voters comprises a spatial pattern which the vote-taker must classify (which the adaptive vote-taker must learn to classify) and for which it must deliver an output decision. The Adaline neuron, utilizing the above-described adaptation procedure, has been applied directly to the job of adaptive vote-taker. Its performance closely approximates the ideal (whose structure is based on sureness information measurements), and is simple to implement physically. The training of the adaptive vote-taker is a continuous process. The "correct" decision is injected at the "desired output" point (Fig. 4). The changes in weight values per computation cycle are made to be exceedingly small. In a practical situation, the time constant of the adaptation process would be of the order of magnitude of the average interval between component failures.

The "correct" decision signal could be supplied externally to permit adapting on check programs. An alternative method would derive this signal from the output decision of vote-taker itself. In Fig. 4, the "desired output point would be connected to the neuron output in a "bootstrap" feedback arrangement. This alternative is the more attractive since it does not require that external signals be supplied to vote-takers dispersed throughout a system, and since adaptation is possible during normal productive system operation. The bootstrap arrangement introduces a stability problem, however. Long chains of random errors could cause the vote-taker to so adapt as to produce incorrect results consistently. This can be prevented by setting the vote weights initially to produce correct results, and by making the adaptation process a very slow one. In system design, the chief problem is to choose a time constant of adaptation long enough to prevent instability and, at the same time, short enough to weed out components as they become defective.

(4) A Pattern-Recognizing Control System

The adaptive networks described in this paper evolved from elementary adaptive control systems. As adaptive pattern-recognizing systems, they may now be used in control systems that can be taught a variety of fairly sophisticated control functions. An example is illustrated in Fig. 10.

The objective of this arrangement is to have the man first learn to control the cart so as to balance the unstable pendulum mounted on it, and then have him teach the adaptive system to do the same thing. As he balances the pendulum, the adaptive network observes both the behavior of the pendulum and the man's reactions, and is adapted

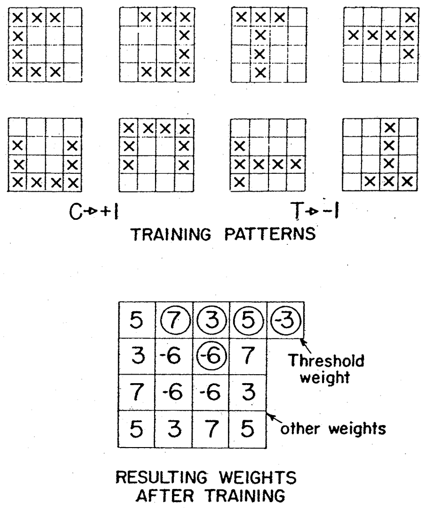


Fig. 9. Training-in insensitivity to rotation of patterns by 90°.

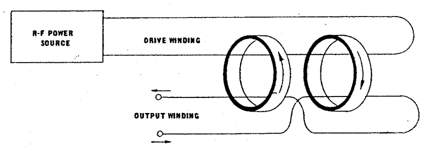


Fig. 10. A magnetic adaptive component.

so as to minimize the error between its output and his reactions. After training, the adaptive network can take over the control of the cart (and thus the pendulum) and control the system.

In a first version, sensors were used to provide information regarding the motion of the pendulum and cart (four-state variables are required). Each of the four-state variables was quantized into a four-bit code, and the resulting 16 binary signals were fed to a single Adaline. After the Adaline was trained its output was used to control the acceleration of the cart (bang-bang control). The single Adaline could almost invariably balance the pendulum while centering the cart over some reference point.

At the present time, a more sophisticated version of the trainable balancer is being made. In this version, the adaptive network will observe the illuminated pendulum with a 7 × 7 retina of photocells, and will balance the pendulum, after training, from the sequences of patterns observed by its artificial "eyeball." This is a far more sophisticated control problem because the input to the adaptive network has not been optimally coded, as it was by the aforementioned four sensors. Hence, this experiment will require a multielement Madaline. When operative, this system should be very instructive in testing the effectiveness of the learning process by comparing the switching functions resulting from adaptation with those that are theoretically optimum for several different performance criteria. The effects of memory capacity and generalization upon performance will be measured.

It is expected that pattern-recognizing control systems will be extremely flexible, and will make possible economical and reliable automation and control of highly complex processes—including processes whose complexities defy mathematical description and analysis.

(5) Construction of Practical Adaptive Networks

The structure of the Adaline neuron and the adaptation procedures used with it have been sufficiently simple to make possible the development of a simple and reliable automatically adapted version of Adaline. The principal technological challenge here has been to provide the weights (the a_i's of Fig. 4) associated with each input to Adaline. The function required is a continuously variable gain with permanent memory, such as might be provided (impractically) with a motor-driven

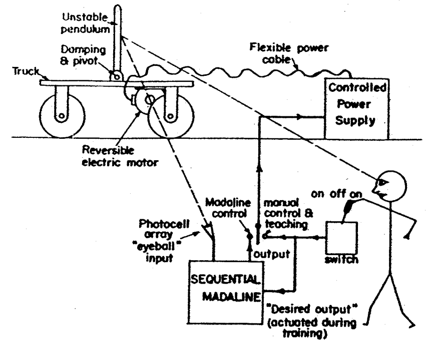


Fig. 12. A trainable balancing system.

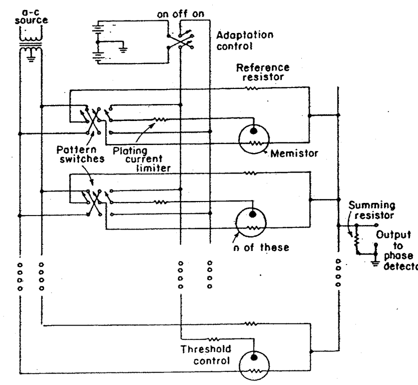


Fig. 13. Schematic diagram of a memistor Adaline.

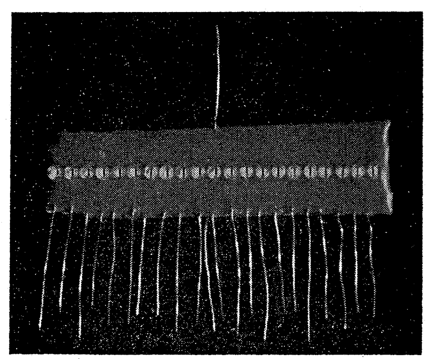


Fig. 14. A partially fabricated sheet of memistors.

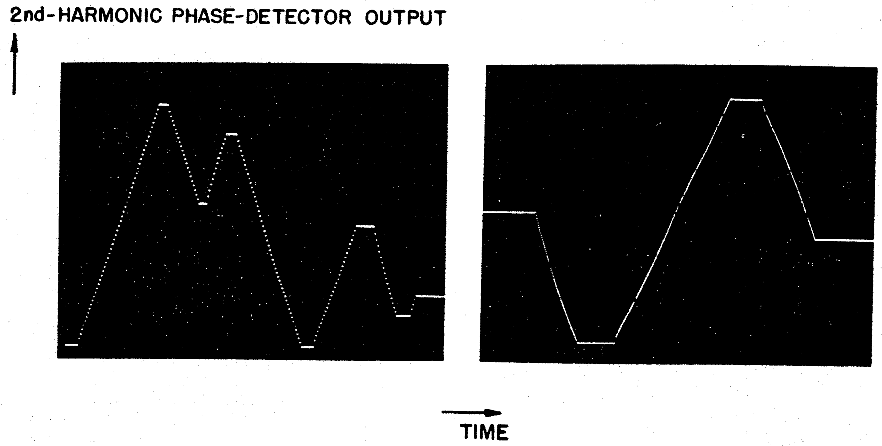


Fig. 11. Training a two-core adaptive component with pulses of direct current.

potentiometer. Because of the need for permanence of memory, purely electronic memory techniques are excluded. However, two components, one based on electrochemical plating and one on analog magnetic memory, have been developed and applied.

The magnetic structure shown in Fig. 11 employs two tape-wound cores, with two windings linking both cores. A 100-kc carrier is applied to the drive winding, which links both cores in the same direction. The sense winding is wound in opposite directions on the two cores, thereby providing a balanced structure. Because the structure is balanced, no 100-kc signal appears in the sense winding. However, a second harmonic, at 200 kc, is developed which is proportional to the average flux stored in the cores. The stored flux can be changed by applying a d.c. adaptation signal to the sense winding. Because the d.c. adaptation signal and the 100-kc carrier are applied together, the stored flux can be varied smoothly, in small increments. The smoothness and linearity of the training characteristic are illustrated in Fig. 12, which show that the amplitude of the second harmonic (actually the output of a second-harmonic phase detector) can be incremented in either direction by the application of short, d.c. adaptation pulses, and that the output remains fixed when the adaptation signal is removed.

The second component is a new electrochemical circuit element called the *memistor* (a resistor with memory), devised by Widrow and Hoff for the realization of automatically adapted Adalines. The memistor provides a single variable-gain element. Each neuron, therefore, employs a number of memistors equal to the number of input lines, plus one for the threshold.

A memistor consists of a conductive substrate with insulated connecting leads, and a metallic anode—all in an electrolytic plating bath. The conductance of the element is reversibly controlled by electroplating. Like the transistor, the memistor is a three-terminal element. The conductance between two of the terminals is controlled by the *time integral* of the current in the third terminal, rather than by its instantaneous value, as in the transistor. Highly reproducible elements have been made which are continuously variable (thousands of possible analog storage levels), and which typically vary in resistance from 50 to 2 ohms, and cover this range in about 15 sec with several tenths of a milliampere of plating current. Adaptation is accomplished by direct current; sensing the neuron logical structure is accomplished nondestructively by passing alternating currents through the array of memistor cells.

A circuit for a memistor Adaline is shown in Fig. 13. Notice the schematic symbol for the three-terminal memistor. This circuit presumes that the neuron input signals are applied by means of switches, and that the overall direction and extent of adaptation are controlled manually. The direction in which each memistor should be adapted (plated or stripped) is determined by the algebraic product of the error signal multiplied by the particular input signal. This product, and hence the direction of adaptation, is affected by the joint action of the adaptation control switch and a gang of each pattern switch (Fig. 13).

In the circuit of Fig. 13, the effect of positive and negative gain values is obtained by balancing the memistor against a fixed resistor in a bridge arrangement. The sensing of the gain is done by applying an a.c. voltage to the memistor, and another a.c. voltage with a 180-deg phase difference to the fixed resistor. The currents are proportional to the conductance and are summed. An individual gain is zero when the memistor conductance equals that of its reference, and an ideal value of reference conductance is the average of the conductance extremes of the memistor. None of the element values or memistor characteristics are critical because of the inherent feedback in the adaptation process.

Fig. 14 presents a photograph of the latest development in memistors, of a type now commercially available. On a single sheet of glass, 21 elements are printed. The actual substrates can be seen in the cell cylinders. Caps and anodes are yet to be installed, and the entire unit is then encapsulated in epoxy. Each cell has a volume of about two drops. Similar cells that were made more than eleven months ago are still working as they did when first constructed, and have been taken through hundreds of thousands of plating-striping cycles with no effect upon electrical characteristics. These cells are essentially insensitive to temperature, shock, and vibration. They have been stored with no deterioration over a temperature range from $-200°C$ to $+100°C$.

Fig. 15 shows a photograph of Madaline I. This machine, constructed at Stanford University, is the largest memistorized machine built to date. It has six Adalines that can be independently adapted, and a total of 102 memistors. The inputs are in parallel, and the present input array is 4×4.

This machine was constructed rapidly during a six-week period. The memistors were not tested before installation in the machine, and some were defective at manufacture. A number of wiring errors were made; some weights were adapting to diverge rather than converge. There were a number of short circuits, open circuits, cold solder joints, etc. This machine worked well when first turned on, and has functioned with very little attention for the past eight months. It took two weeks of experimentation before suspicions were aroused and the weights were checked. Twenty-five percent of them were not adapting. Yet the machine was able to adapt around its own internal flaws and to be trained to make very complex pattern discriminations. These errors were corrected, and the machine's capacity increased accordingly.

Madaline I has just been put under the control of an IBM 1620 com-

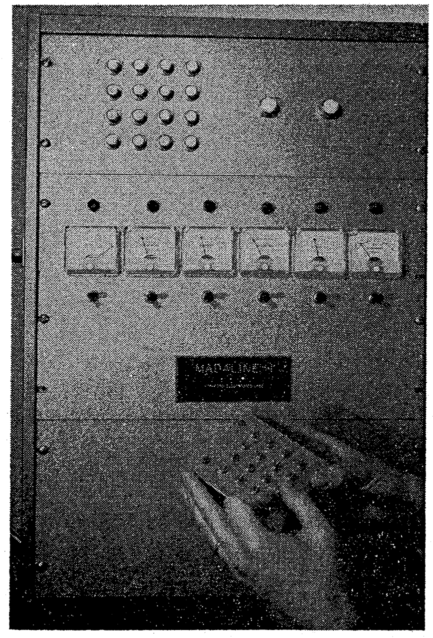


Fig. 15. Madaline I.

puter. The computer stores the patterns and desired responses, and controls the training of the neurons. It tabulates the number of patterns seen and adaptations made, and error probability as the learning process progresses. After the current task of expanding Madaline I to six 49-input Adalines is completed, it, together with the 1620 computer, will be three times faster than the digital simulation of the same structure on the 1620 alone. A still larger Madaline is being planned which will contain 1,500 adaptive weights. When connected to the 1620 computer, it will be faster at neuron simulation than an IBM 7090. On this scale, a neuron simulation facility consisting of a small computer and memistorized neurons is ten times cheaper than an all-digital simulation facility.

The fundamental objective in connecting adaptive neurons to a computer is to develop a new type of computer, one as different from the digital as the digital is from the analog. This new type of machine might be called the *adaptive computer*. The basic "flip-flop" for it is Adaline. The adaptive computer is taught rather than programed to solve problems. The job of the "programer" is to establish suitable training examples. This machine will be taught by men (so that it will solve the problems of men) in the environment and with the language of men, not with a machine language. The learning experience derived from human teachers will provide reasonable initial conditions, upon which the machine could subsequently improve from its own systematic experimentation and experience-gathering.

References

[1] Suran, J. J., *Use of Passive Redundancy in Electronic Systems*, IRE Trans. on Military Electronics, Vol. MIL-5, July 1961.

[2] Phister, Montgomery, Jr., *Logical Design of Digital Computers*, Chap. 10; John Wiley & Sons, Inc., New York, 1958.

[3] Shannon, C. E., and Moore, E. F., *Reliable Circuits Using Less Reliable Relays*, J. Franklin Inst., Vol. 262, Sept.–Oct. 1956.

[4] Von Neumann, J., *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components*, Automata Studies; Princeton Univ. Press, Princeton, N.J., 1956.

[5] Pierce, W. H., *Improving Reliability of Digital Systems by Redundancy and Adaption*, TR No. 1552-3, Stanford Electronics Labs., Stanford, Calif., July 17, 1961.

[6] Widrow, B., and Hoff, M. E., Jr., *Adaptive Switching Circuits*, IRE Wescon Convention Record, Pt. 4, Aug. 23, 1960.

[7] McNaughton, R., *Unate Truth Functions*, TR No. 4, Applied Math. & Statistics Lab., Stanford, Calif., Oct. 21, 1957.

[8] Mattson, R. L., *A Self-Organizing Logical System*, Proc. of Eastern Joint Computer Conf., 1959.

[9] Widrow, B., *Adaptive Sampled-Data Systems—A Statistical Theory of Adaption*, IRE Wescon Convention Record, Pt. 4, 1959.

[10] Ridgway, W. C., III, *An Adaptive Logic System With Generalizing Properties*, TR No. 1556-1, Stanford Electronics Labs., Stanford, Calif., Apr. 1962.