SIMULATION OF SELF-ORGANIZING SYSTEMS BY DIGITAL COMPUTER *

B. G. Farley and W. A. Clark
Lincoln Laboratory, Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

A general discussion of ideas and definitions relating to self-organizing systems and their synthesis is given, together with remarks concerning their simulation by digital computer. Synthesis and simulation of an actual system is then described. This system, initially randomly organized within wide limits, organizes itself to perform a simple prescribed task.

INTRODUCTION

Information systems whose response to a given class of inputs changes with time in accordance with specified criteria which are chosen to correspond roughly to the "self-organizing" concept have been the subject of considerable interest.[7,12] Several mechanisms have been constructed or described which are "self-organizing" to some extent,[1,6,8,11] and some work has been published on computer-programmed learning, such as that by Oettinger.[5] Recently, McKay has communicated ideas related to some of those to be discussed here.[4]

The work to be described was undertaken in an attempt to clarify certain ideas related to such systems, and to try to gain some insight into their synthesis by simulation of specific systems using a digital computer. Although the work is in an early stage, it is believed that results so far have exhibited some very interesting properties of a particular system, and have demonstrated the usefulness of computer simulation methods in studies of this kind where systems are likely to be so complex that analytical solutions are difficult or impossible, or do not furnish much information until leads are suggested by actual experience.

The work will be presented in three parts. First a general discussion will be given in which definitions will be made. Second, the definitions will be applied to an experimental system. Third, the details of a self-organizing system and a description of computer techniques used in its simulation will be given.

General Considerations and Definitions

In order to make our ideas and definitions precise, and at the same time as general as possible, it is convenient to introduce a mathematical framework to aid in discussion.

We will deal first with a general system as shown in Figure 1. Inputs p_j from the left are transformed into outputs q_j on the right. As indicated in Figure 1, both input and output lines may be multiple. In what follows, the symbols p_j and q_j will refer to specific, complete configurations on these multiple lines, finite or infinite in time. No loss of generality will result if all signals are reduced to a binary equivalent. As an example, then, if there are three input lines, a certain input might be defined as

$$p_3 = \begin{cases} 0110001011001 \\ 1001101011001 \\ 0110101000000 \end{cases} \tag{1}$$

time increasing to the right. Such a configuration will be called a time-channel pattern.

The transformation T will be allowed to change with time, and we are interested in this change in so far as it exhibits organizing properties with respect to T. To define properties of this type we will fix our attention on a particular class C of inputs p_j, $j=1,2,\ldots,n$ and their corresponding outputs q_j. Each member of this class will usually be finite in length.

We may then break the transformation T down into a class of transforms

$$T = \left\{T_1, T_2, \ldots, T_n\right\} \tag{2}$$

*The research in this document was supported jointly by the Army, Navy, and Air Force under contract with the Massachusetts Institute of Technology.

where the set of equations

$$\begin{aligned} T_1 p_1 &= q_1 \\ &\cdots \\ T_n p_n &= q_n \end{aligned} \tag{3}$$

serves to define $T_1 \ldots T_n$ If the system contains sources of noise or produces spontaneous outputs, the transformations $T_1 \ldots T_n$ will be defined statistically as averages over an ensemble of identical systems started in the same initial state.

Now, in order to discuss one or more properties of such a system dependent on time, it is only necessary to choose a measure m specifying the properties in question and apply it to T at succeeding times. In many cases, the measure $m(T)$ will of course depend upon only one, or a few, of the T_j's.

We will consider that the instrument of time change of T is within the system. As an aid to visualization, Fig. 2 shows the system broken down into two components, one of which contributes primarily to the transformation T itself, and the other, called the modifier, has the primary function of producing changes in T. The double line between the modifier and T represents the agency of the modification, while single lines show information paths. While a sharp dichotomy of function between T and modifier has been indicated, it is not intended to exclude systems in which the modifier contributes to the transformation or modifies itself. We will consider everything outside the dashed lines as "environment," although it should be noted that the exact path of such boundaries is arbitrary.

It may be of some interest to suggest as an illustrative example how the above model as described might be used to describe situations which approximate psychological definitions of "learning."

Learning behavior by an organism may be defined for the purposes of psychology as a positive change in the proficiency of performance of one or more tasks as a result of prescribed experience.[9]

In terms of our model, we may describe this as follows. A number of input patterns are chosen, and presented to the organism in prescribed orders and times to provide the required "experience." One or more of these inputs are designated as performance tests or tasks, and a suitable proficiency measure, such as a test score, is constructed. This score corresponds to the measure we have attached to the general transformation. If the measure increases as a result of presentation of "experience" inputs, and does not increase otherwise, the organism is said to learn.

The provision that the measure should increase only as a result of presentation of "experience" rules out as learning systems those in which the modifier operates to increase a measure without information inputs. Control experiments may of course be required to rule out such cases.

It should be noted that learning thus defined is relative to the input class and measure chosen, and the experience prescribed. By varying these parameters, various kinds of learning may be defined. For example, transfer learning requires altered experiences or measures of new performance (perhaps with special control); learning with relatively short performance inputs is called conditioning, while that with long performance inputs is called serial learning. More precise definitions would require close examination of the variable parameters. This task is complicated for the psychologist by the fact that he is dealing with organisms no two of which are alike.

Some competing theoretical interpretations of learning may also be referred to the model. For example, reinforcement and non-reinforcement theories make different assumptions as to the nature of the modifier organization. "Perception" theorists make use of "perceptual systems or fields" which are not explicitly represented in our model as presented here.[9]

No matter how complex the organization of a system such as we have been discussing, it can always be simulated as closely as desired by a digital computer as long as its rules of organization are known. This possibility is indicated for example, by the work of Turing.[10] This means that the action of any system can be studied even though it is too complex for mathematical analysis. Furthermore, the computer offers unparalleled flexibility in such work, since any part of a simulated system may be quickly and easily modified to judge the effect of the change. There is of course the disadvantage that present computer simulation takes pace serially in time, so that even with very fast computers considerable time may be required to simulate highly complex systems. Balanced against this disadvantage, however, is the fact that the initial programming for simulation in general requires a great deal less time than actual construction of an analogue device even if this is feasible, so that for a very wide class of problems the net advantage in both time and cost lies on the side of a computer simulation method, and for an additional large class this method is the only feasible one, at least until the system is reasonably well understood.

The work to be described was undertaken partly to examine the problems encountered in such simulation. Furthermore, it was desired to answer two questions: (1) Can a transformation, initially organized at random between rather wide limits, be provided with a modifier which will cause it to become organized, as a result of experience, to perform a prescribed task? (2) Can such a system be generalized to organize itself to perform any of a rather wide class of tasks? The work to be described is still in an early stage, but has resulted in the synthesis of a system which it is believed fulfills the requirements of the first question.

Application to an Experimental System

In seeking to synthesize systems along the lines discussed above, it is natural first to choose a transformation with promising transforming and modifiability possibilities and then try to discover suitable modifiers. Preliminary investigation showed that transformations composed of interconnected active non-linear elements with definite thresholds as indicated in Fig. 3 have interesting transforming properties. For example, such a net of elements can change a time-channel pattern into a space pattern of active elements, and if it is complex enough, can do this uniquely for a given class of patterns. Furthermore, enough variable parameters are available in the net to give it useful modifiability properties. Networks resembling those under discussion exist naturally, and have a great intrinsic interest, namely networks of nerve fibers or neurons.[3] It was therefore decided to use non-linear elements possessing many of the known properties of neuron nets as an experimental transformation.[3] The details of the net and associated modifier will be presented later, but first the simple task chosen for performance, the measure of proficiency used, and the prescribed experience, will be described in terms of the framework already discussed.

First a randomly connected net is arbitrarily divided into four groups of elements designated as groups I_a, I_b, $O(+)$, and $O(-)$. These symbols stand for input groups "a", "b", and output groups "+", and "-", respectively.

Two input patterns, p_1 and p_2 are considered. The first, p_1, may be represented by the following scheme,

$$p_1 = \begin{cases} \left.\begin{matrix} \ldots 001001001 00\ldots \\ \ldots 00100100100\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \end{matrix}\right\} I_a \\ \left.\begin{matrix} \ldots 00000000000\ldots \\ \ldots 00000000000\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \end{matrix}\right\} I_b \end{cases} \tag{4}$$

which indicates that the same periodic input is applied to every element of I_a, and that no input is applied to I_b. The input p_2 is identical except that the roles of I_a and I_b are reversed. When p_1 is applied, the transformation called T_1 is active, and T_2 is active when p_2 is applied, in accord with equation (3). In order to define a proficiency measure, we proceed as follows: Let $n(+)$ be the number of elements active during a given time interval in group $O(+)$, and $n(-)$ the number active during the same interval in group $O(-)$.

The measure $m(T)$ will be composed of two components, m_1 and m_2.

$$m(T) = \{m_1,\ m_2\} \tag{5}$$

where

$$\begin{aligned} m_1 &\equiv m_1(T_1) = \overline{n(+) - n(-)} \\ m_2 &\equiv m_2(T_2) = \overline{n(-) - n(+)} \end{aligned} \tag{6}$$

and the bar denotes a time average over a fixed interval.

Note that m_1 is defined above only when T_1 is active, and similarly for m_2 and T_2. Organization will be said to occur if both m_1 and m_2 increase.

In other words, we may consider an output formed by the accumulated difference of the numbers of cells active in $O(+)$ and $O(-)$. Presentation of experience will be externally arranged so that p_1 is applied whenever the output is positive, ($O(+)$ predominates) and p_2 whenever the output is negative, ($O(-)$ predominates). If the output remains near zero for a specified length of time, it is externally "forced" from zero by adding to the output difference in alternately positive and negative directions. Thus the whole mechanism is similar in some respects to a servo which must learn to return to zero when displaced, training experience being given alternately on either side of zero, and increasing organization being manifested by an increasing rate of return. The patterns p_1 and p_2 provided by the environment may be said to enable the mechanism to "sense" the position of its output.

The modifier which causes the measures m_1 and m_2 to increase was determined largely empirically. It operates on various parameters of the net in a way to be described later. Information for the operations of the modifier is generated internally in this simple case in a manner which essentially computes m_1 and m_2. However, it should be mentioned that in the general case this may not necessarily be true. That is, the modifier may use information related to the organization measure, but computed in some entirely different manner.

Details of Experimental System and Simulation Program

The general properties of the particular transformation and associated modifier with which the initial simulation work has dealt have been presented. This description will now be expanded and related to the computer simulation techniques which were developed for the Memory Test Computer of the Lincoln Laboratory of M.I.T. A note on the characteristics of the computer may be of general interest: MTC is a 16 bit parallel machine with a coincident current magnetic-core memory of 4096 words and an operating speed of about 90,000 single-address add-type instructions per second. Its principal input device is a Ferranti Photoelectric Reader for punched paper tape and output equipment includes a standard flexowriter and several cathode ray tubes for displays which may be photographed.

The transformation system has been described as a network of non-linear elements in which the pathways or connections are randomly established. In this and other parts of the program random processes play an important part and should be discussed in more detail. MTC does not have access to a random element, but there exist many accredited computation routines which generate number sequences in which the values of the terms are distributed in a nearly statistically homogeneous manner. The pseudo-random number generator routine which was used develops the n^{th} terms, R_n, by means of the recursion relation

$$R_n = R_{n-1} + R_{n-k} \quad \text{(Sum modulo p)} \tag{7}$$

The series initially is "primed" with k terms chosen from a table of random numbers.

To connect network elements at random a matrix P_{ij} expressing the probability that i connects j is established for the class of networks under consideration. In the systems to be discussed, the simple case $P_{ij} = K$, constant for all i, j was chosen, but more generally the connection probability might depend on i, j or any particular characteristics of network elements i and j. For each pair of network elements a pseudo-random number in the interval ab is then generated and a test is made to determine whether the number lies also in a subinterval ar of ab where r is so chosen that the ratio of (r-a) to (b-a) is the probability P_{ij}. Since the pseudo-random numbers are uniformly distributed in the interval, this test yields positive results with a mean relative frequency equal to the required probability. For each positive test result, a connection is established and in this way a specific connection matrix, ($c_{ij} = 1$ if i connects j, 0 otherwise), is set up for the given network. K will be called the connectivity of the net.

With each connection there is associated a sixteen-state weight, w_{ij} which determines the excitation value on j of activity transmitted from i via this connection (see fig. 3). These weights may in general be drawn from a distribution in the manner discussed above, although in the example presented later these weights were chosen equal and set initially at mid-value.

For each element in the network, one row of the connection matrix (representing pathways from the element) and a list of associated weights are stored in the computer memory. This requires breaking the matrix row into 16-bit words and also packing four of the 4-bit weights into one word for storage economy, and much of the computing time is consumed in the unscrambling and repacking of these words during the simulated operation of the network.

Similarly, with each network element there is stored a list of characteristics such as threshold, time constants, etc. selected from appropriate distribution functions, and addresses and counters required by the simulation process. These quantities occupy another six 16-bit words. The total storage requirement, however, is determined largely by the connection matrix and associated weights, since for these the required capacity increases with the square of the number of network elements. The 4096 registers of the MTC memory limit the size to a network of about 128 elements with connectivity of 0.4. The time required to generate such a net is approximately ten seconds, or about 900,000 operations. The complete simulation program occupies about 1500 registers; the remainder of the storage is occupied by the characteristics of the network.

In order to elaborate on the characteristics of the network elements it is necessary to discuss more completely the transient behavior of the element during excitation. This transient state of activity occurs whenever the excitation level exceeds the threshold of the element. After a small time delay, the element transmits by simultaneously increasing the excitation level of all other elements to which it is connected as indicated by its associated row in the connection matrix. At the beginning of this delay interval, the threshold rises to a value which is large enough to prevent a second activation during the interval. At the end of this interval, suggested by the refractory period in neurons, the element recovers sensitivity as it threshold decays exponentially to a minimum value characteristic of the element, measured relative to an adjustable bias level for the network as a whole. The threshold function, $h_j(t)$, for the j^{th} element may thus be represented as effectively infinite during the refractory interval and

$$h_j(t) = h_{max}\exp(-a_j t) + h_{jmin} + h_{bias}(t) \quad (9)$$

otherwise, where a_j is the threshold decay constant. The comparison of excitation with threshold occurs in the presence of gaussian noise such that a high level of excitation increases the probability that an element will "fire" but will not in general completely determine the instant of firing. The behavior of the network becomes completely determinate as the mean-square amplitude of the noise, which, like the bias level, is controlled by the modifying sub-system, is reduced to zero. The gaussian distribution is approximated, as suggested by the central limit theorem, by averaging a set of four pseudo-random terms for each term of the "gaussian" set.

When several elements simultaneously transmit to the same element, the change in excitation of the affected element is chosen to be the sum of the weights of the active connections, although a more complicated function of the weights might be used. In addition, the total excitation level at the affected element decays exponentially with a time-constant characteristic of the element. Thus, activity pulses arriving within a small time-interval of one another partially combine in excitation value in a manner related to the observed temporal summation effects in neurons. The change in excitation, $\Delta s_j(t)$, at the j^{th} element at time t may then be written

$$\Delta s_j(t) = -b_j\, s_j(t-1) + \sum_i w_{ij} \quad (9)$$

where the summation extends only over elements which transmitted at t-1 excluding, in the model chosen, i=j, and b_j is the excitation decay constant.

The characteristics stored in the computer memory for the j^{th} network element can now be enumerated in summary:

(1) Type of element, i.e. number of the group I_a, O(-) etc. to which the element is assigned.

(2) Time delay, which determines the refractory period, and also, in the simple model chosen, the delay between firing and transmitting (equal for all pathways from the transmitting element).

(3) Minimum threshold, h_{jmin}

(4) Threshold decay constant, a_j

(5) Excitation decay constant, b_j

(6) Connection Matrix row, c_{jk}, k=1,2,..,n where n is the number of elements in the network.

(7) Those connection weights, w_{jk}, for which $c_{jk}=1$

It should be pointed out that as yet there has been no systematic evaluation of the effects of varying thresholds, decay constants, and time delays. Their inclusion in the set of characteristics does, however, illustrate the degree of complexity of the model being simulated.

In the simulation program the time variable is quantized into equal intervals of about one-eighth of a refractory period. This time parameter is, in effect, frozen until the program has scanned through storage, calculating values of threshold, excitation, etc. for each element, after which it is advanced to the next larger value. The real time consumed per "time" interval in carrying out these calculations for a net of 128 elements with connectivity of 0.4 is about one second, varying from interval to interval according to the amount of activity within the network.

Activity is introduced into the network by increasing by a large fixed value the excitation level of those input elements, and at those times, indicated by the presence of "ones" in an input pattern similar to the p_1 of eq. (4). The output, as described earlier, is formed simply by counting the number of transmitting elements in the output groups O(+) and O(-) during each time interval. The difference of these numbers, $n_t(+) - n_t(-)$, defines the changes in the output N_t of the net so that

$$N_{t+1} = N_t + n_t(t) - n_t(-) \quad (10)$$

The computer program is arranged to plot N_t against t directly on one of the display scopes and a time exposure photograph records the trace. A typical output record appears in fig. 5. In order to automatize the process of presenting input patterns, the simulated external system is so arranged that $N_t > +N'$ results in pattern p_1 and $N_t < -N'$ produces p_2 where N' is a small positive number. If N_t remains in the null interval between $-N'$ and $+N'$ for a specified period (chosen long enough to allow residual activity to attenuate) the program displaces N_t to some value $+N'' > N'$ with alternate trial displacements to $-N''$. These displacements will be seen as the "discontinuities in fig. 5.

The action of the modifying system is best described by means of the flow diagram of fig. 4. If a "contributive connection" is defined as any active connection to a "fired" element which may have contributed to the firing of the element during an immediately previous fixed time, the modifier increases the weights of contributive connections when the magnitude of the output has just decreased and decreases these weights if the magnitude of output has just increased, subject to upper and lower bounds of weight value. Note that weights are changed without regard to their individual influence on the output, and improvement in performance results from what might be termed "statistical cooperation." In addition, the modifier manipulates the threshold bias level and the noise level within the net, the former by gradually lowering bias until activity starts (principally to prevent self-sustained activity, which is difficult to control) and the latter to allow noise-initiated activity to scan, in effect, new activity modes of the network when required. Bias control of this sort may be considered use of a "field" parameter, in contrast to use of local cell parameters.

A Small Network Example

An example of an eight element network of 0.75 connectivity will now be given. To simplify the network for illustrative purposes, the elements are divided into four equal groups and numbered so that elements in the same group are represented by successive rows in the connection matrix. In this example, a_j=0.25, b_j=0.50 for all j, and the refractory delays were all equal to 2 time units.

Fig. 5 shows the history of the output of this network during the organization process requiring approximately 15 minutes of computer time. (The graph is redrawn from a set of photographs which were unsuitable for reproduction). Up to the point marked "modifier activated", the behavior of the unaltered transformation is seen to be slowly divergent for both positive and negative test displacements. Figs. 6a through 6d show the weight matrix sampled at the times indicated by points labeled "a" through "d" in fig. 5. The numbers appearing in these matrices are in octonary form and will be seen to change substantially during the organizing process.

It will be noted that the changes primarily affecting the output occur in the enclosed boxes; weights in boxes I_a O(+) and I_b O(-) tend to increase while those in boxes I_a O(-) and I_b O(+) tend to decrease. It can be seen the return to zero of the output gradually improves from a condition of divergence to increasingly rapid convergence as the matrix changes progress.

A total of perhaps 30 randomly organized nets of this type with various connectivities have actually been tried, the largest of which contained 64 elements with K=0.75. All but 3 or 4 have been organized successfully by the modifier, the failure being due to lack of essential connections or other special properties sometimes resulting from the wide variability of the random process.

It might also be of interest to note that exploratory experiments have been made to examine the effect of damage on these nets after organization. Indications are that arbitrary destruction of at least 10% of the elements may be sustained without impairment of performance.

Conclusion

We have now described a general formulation of the self-organizing concept, and a synthetic example of a system which organizes itself to perform a simple task.

Although the experimental system was composed of elements having properties similar in many respects to the known properties of neurons, it is not claimed at this stage that the results are of neurophysiological significance. However, it is believed that the results do show the great usefulness of computer simulation methods in this and other fields where systems of great complexity are encountered. Not only will simulation methods produce specific knowledge, but it is believed that they should also eventually yield enough information about given types of systems to make more general formulations possible. For example, enough experience has not yet been gained about the present experimental system to understand what features are necessary under given conditions, but it is believed such information can be elicited by an extension of the present methods.

As mentioned earlier, the gradual organization of the system to utilize the patterns p_1 and p_2 to change an output in opposite directions implies a primitive "recognition" of these patterns. It is also found experimentally that after organization other patterns also have effects like p_1 or p_2. In other words patterns are classified together by such a transformation. It is to be hoped that, using a more complex modifier, this type of behavior can also be organized and controlled, leading to systems which effect classifications and generalizations. Success in this respect should make possible systems which can organize themselves to perform in an environment presenting a rather wide variety of tasks.

Acknowledgment

The authors wish to express their appreciation to F. A. Webster for many valuable discussions, and also to those responsible for the operation of the Memory Test Computer for their very helpful cooperation.

REFERENCES

1. Ashby, W. R. Design for a Brain, Wiley, 1952

2. Brazier, M. A. B. The Electrical Activity of the Nervous System, MacMillan, 1951

3. Hebb, D. O., The Organization of Behavior, Wiley, 1949

4. McKay, D. M., (to be published) Oral communication at conference on Human Communication and Control, M.I.T., June 1954.

5. Oettinger, A. G., "Programming a Digital Computer to Learn" Phil. Mag. 43, 1243-1263

6. Shannon, C. E., "Presentation of Maze-solving Machine", Transactions of the Eighth Cybernetics Conference of the Macy Foundation, 1952, 173-180.

7. Shannon, C. E., "Computers and Automata", Proc. I.R.E., 41, 1234-1241 (Oct. 1953)

8. Shimbel, A., "Contributions to the Mathematical Biophyics of the Central Nervous System, with Special Reference to Learning," Bull. Math. Biophysics, 12, 241-274

9. Stevens, S. S. (ed.) Handbook of Experimental Psychology, Wiley, 1951, Chap. 16

10. Turing, A. M., "On Computable Numbers, with an Application to the Entscheidungspreblem" Proc. Lond. Math. Soc., 24, 230-265, (1936)

11. Walter, W. G., The Living Brain, Norton, 1953

12. Wilkes, M. V., "Can Machines Think", Proc. I.R.E., 41, 1230-1234 (Oct. 1953)

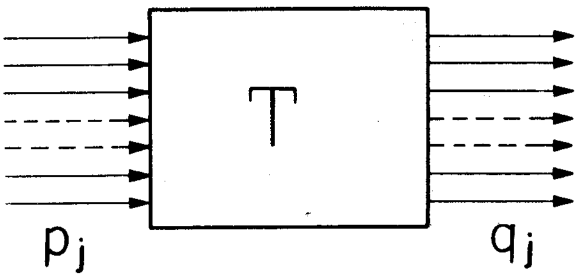


Fig. 1 - General transformation.

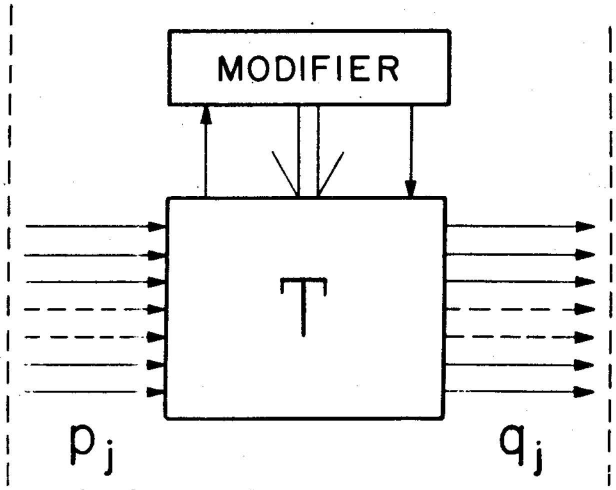


Fig. 2 - General self-organizing system.

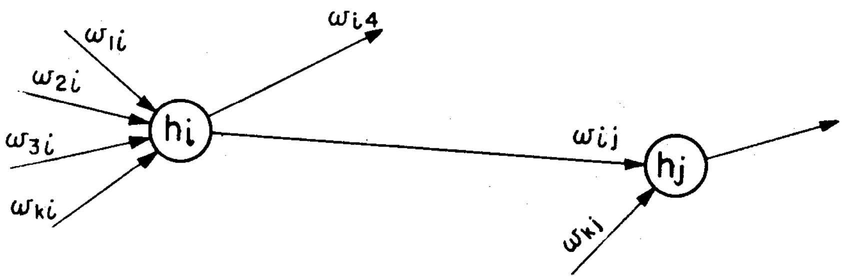


Fig. 3 - Typical section of network showing weights, w, and thresholds, h, associated with nonlinear elements i and j.

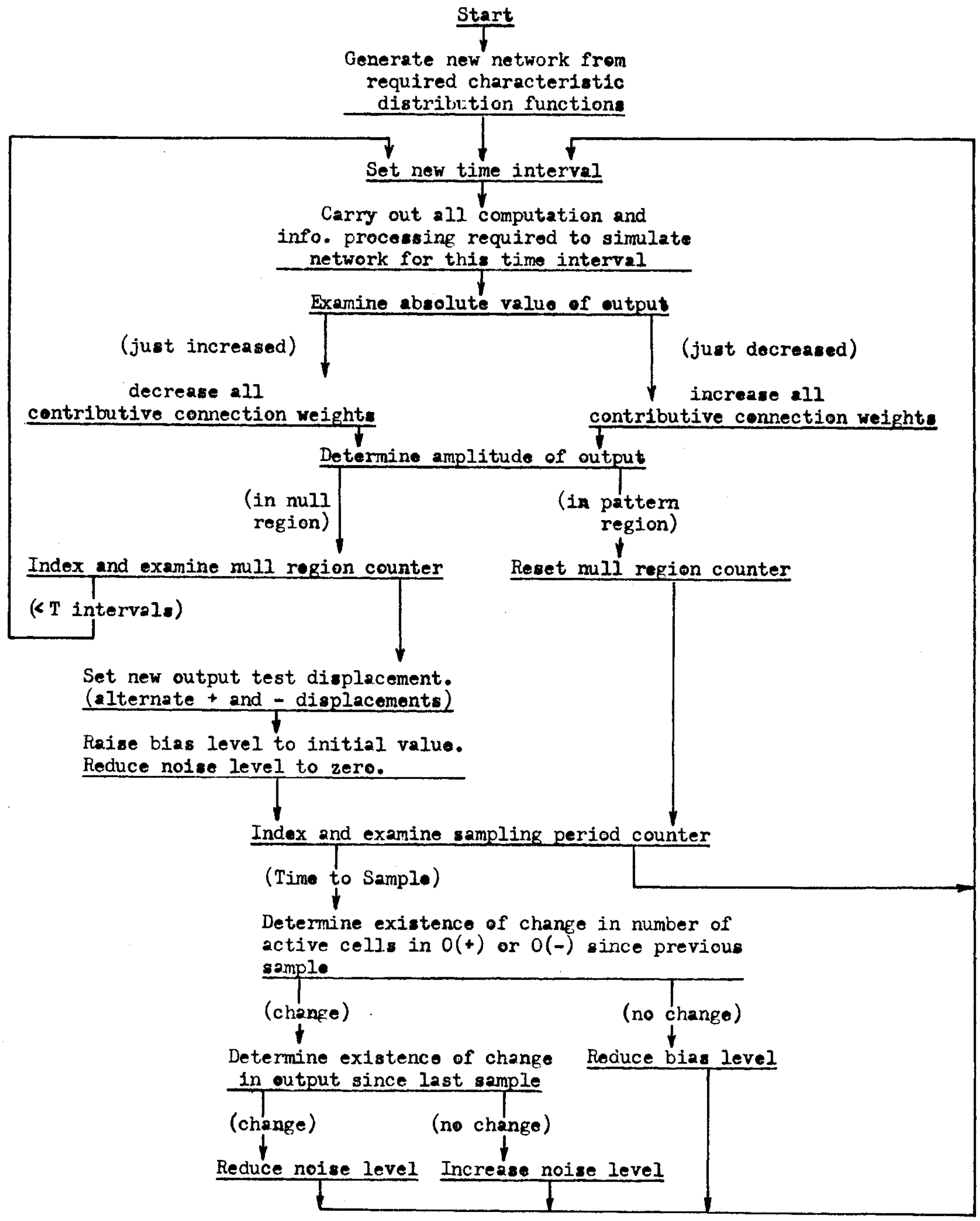
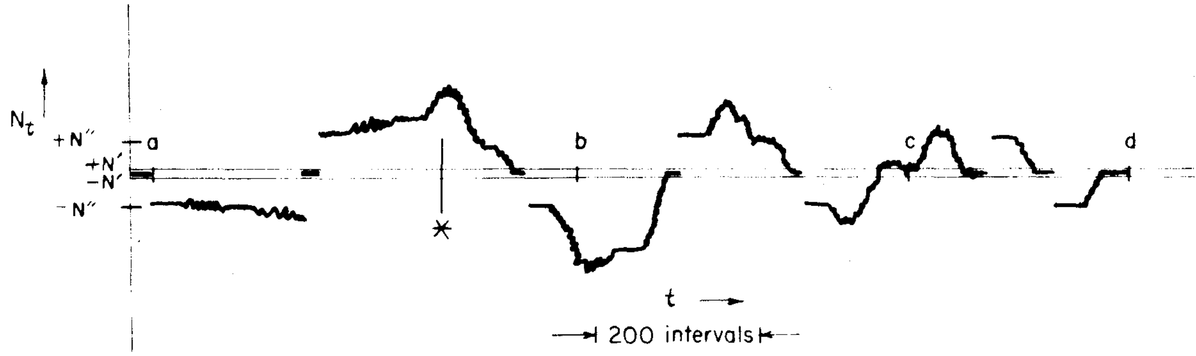


Fig. 4 - Simplified computer simulation flow diagram emphasizing modifier.



✱ = modifier activated

Fig. 5 - Output record of an 8-element network; K = 0.75.

	I_a		$O(-)$		$O(+)$		I_b	
I_a	07	07	07	07	07	07	07	07
		07	07			07	07	07
$O(-)$			07	07	07	07		07
	07	07	07	07	07	07	07	07
$O(+)$			07	07		07	07	07
	07		07		07	07	07	07
I_b	07			07		07	07	07
	07	07			07		07	

(a)

	I_a		$O(-)$		$O(+)$		I_B	
I_a	07	10	07	11	04	11	11	11
		07	03			02	02	07
$O(-)$			07	06	01	01		07
	07	06	06	07	01	01	01	02
$O(+)$			14	17		10	12	07
	04		10		01	07	02	05
I_b	10			14		06	07	07
	05	12			01		06	

(b)

	I_a		$O(-)$		$O(+)$		I_b	
I_a	07	10	02	01	15	07	01	16
		07	01			17	01	07
$O(-)$			07	04	11	14		07
	16	04	02	07	07	10	01	10
$O(+)$			03	11		03	02	07
	13		01		11	07	01	12
I_b	17			11		16	07	07
	10	01			05		01	

(c)

	I_a		$O(-)$		$O(+)$		I_b	
I_a	07	03	02	03	02	03	01	14
		07	01			16	01	07
$O(-)$			07	04	11	12		07
	15	01	03	07	01	02	01	06
$O(+)$			03	12		04	03	07
	15		01		04	07	01	11
I_b	17			16		08	07	07
	17	01			02		01	

(d)

Fig. 6 - The connection weight matrix for the 8-element network sampled at points labelled "a" through "d" in Fig. 5.