

THE UNIVERSITY OF MICHIGAN  
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS  
Computer and Communication Sciences Department

ALTERNATIVE FORMALISMS FOR  
BIO- AND ECO-SYSTEM MODELLING

Bernard P. Zeigler  
Andrew G. Barto

February 1975

Technical Report No. 168

with assistance from

National Science Foundation  
Grant No. DCR71-01997  
Washington, D.C.



Biologists and ecologists have traditionally turned to the differential equation as the means of formulating hypotheses about the operation of the natural world. Indeed, this mathematical tool has helped to express quantitatively such key relationships as are involved in, for example, the allometric growth of organisms, the genetic adaptation of populations, the biochemical activities of cells, and the energy transfers among species linked in an ecological web. And there is little doubt that the differential equation will continue to play a significant role in the arsenal of the biological and ecological modeller in the future.

However the very success and prominence of the differential equation has also worked against the development and application of other formalisms for model description - formalisms which have a more recent origin and which could open up new possibilities for our understanding of natural phenomena.

Roughly, a formalism is a vigorous means, mathematical in appearance, of specifying a model. The differential equation invented by Newton is one such formalism, so is the difference equation, which is its discrete-time counterpart. It is important to understand the use of modelling formalisms since each carries with it a "world view", i.e. makes certain hypotheses about the real world easily and naturally expressible, while making other concepts difficult or impossible to capture. The world view of the differential equation is that of "rates of change" - all activities are formulated through laws governing the rates at which their state variables are altered. In this paper, we shall discuss formalisms, and their world views, of more recent origin.

Alternatives to the differential equation have been suggested for biological phenomena for some time, at least since the 1940's. That period saw the birth of the relational biology concepts of Rachevsky, the

cellular automaton description of self reproduction by von Neumann and the logical modelling of neuron behavior of McCulloch. However, despite continued development, the descriptive formalisms underlying these models have not yet entered the mainstream of biological thinking, and certainly have had little impact on ecology.

Recently there have been some signs of change. Most conspicuously, the use of formal grammars to model the development of form in plants and animals has seen a dramatic surge of interest, chiefly due to the initiative of the biologist Aristid Lindenmeyer.

It is the underlying belief of this paper that such developments will become increasingly significant in the near future. The reason for this, we think, lies in the explosive use of the digital computer. The simulation capabilities of the computer have made possible the contemplation of models of scales and forms previously undreamt of. Our conception of what models are has been radically altered in the process.

In this paper, we shall briefly discuss two classes of model description formalisms which we, as computer scientists, have recently become interested in. The formalisms are varieties of the "discrete event" modelling approach first developed in connection with industrial process simulation, and the "uniform cellular space", originally due to von Neumann. We shall provide evidence of biological and ecological applicability, but the early stage of development allows us only to hope to be suggestive rather than definitive.

A general discussion of the nature of model description in the context of computer simulation will be found in Zeigler (1975). For now, it suffices to say that we regard a model as basically a set of instructions for generating dynamical behavior, i.e., input-output time-segment pairs, which can be matched with real system observations similar in form. These instructions

may be expressed via differential equation, automata theoretic, discrete event, or other constructs. But the essential thing to realize is that the model structure, i.e., its set of instructions, has a life of its own, independent of any particular program realization of the model. Formally speaking, a model is a means for specifying a dynamical system. A model description formalism provides the standardized means with which a specific system from a larger class can be so specified (e.g., a particular differential equation specifies a particular member of the class of smooth dynamical systems.)

#### Discrete Event Motivation

An illustrative use of discrete event modelling in ecology is given by Sigal and Pritsher (1974). Their model consists of two parts (Fig. 1). The first part is a lake ecosystem, classically formulated, i.e., employing Holling predator-prey relationships to express the rates of energy transfer via differential equations. The second part models ecosystem control policies. It is formulated in discrete event terms to realize such events as stocking the lake with a designated species periodically, replenishing the lake with a species when the level of another species surpasses a designated level, and spraying a species at preset times. Each of these activities occurs instantaneously at its scheduled time and causes an immediate resetting of appropriate state variables in the ecosystem model.

The example illustrates the potential use of simulation for trying out ecosystem control policies before actual implementation. It also illustrates the use of hybrid models employing both differential equation and discrete event components. Both of these implications are likely to become increasingly significant in the future. However, the example does not go very far in illustrating the use of discrete event concepts in modelling biological and ecological phenomena per se.

True, one might infer from the example that regularly recurring natural phenomena such as seasonal and diurnal climatic changes might be conveniently scheduled in a discrete event model. But the potential of the discrete event philosophy for capturing intrinsic and important eco- and bio-system relationships goes much deeper than this. We refer here to the programming of activities, their timing, release and inhibition, and the subtle phasic interrelations among activity cycles which can be found at all levels of the biosphere from the biochemical factory of a cell to the life cycles of plants and animals.

We shall provide an illustration of this potential in a moment after we review the discrete event philosophy which comes next.

#### Discrete Event Philosophy

We shall provide an informal view of the nature of discrete event modelling. The constructs given correspond to standard simulation language (Simscrip, Simula, GPSS) features and have been formalized (Zeigler (1975)).

A model consists of a set of interacting components. In a discrete event model, each component can be thought of as consisting of a set of processes. We can think of a process as akin to a computer program - it lays down a sequence of activities to be carried out. The execution of activities, however, is not automatic, as time and external conditions may intervene to space out, inhibit, or change the course of the activations.

An activity can schedule a succeeding activity to occur at a definite time in the future. In this case, which we call unconditional scheduling, the successor activity will not take place until the designated time is reached and then it will be executed instantaneously at that time. Or an activity can schedule a succeeding activity to occur after some definite time, and as soon as the appropriate conditions subsequently arise in the

model. In this case, of conditional scheduling, the successor activity will not take place until the designated time is reached. At that time, and at subsequent times at which "events" occur in the model, the activating conditions will be tested. When these conditions are finally satisfied, the activity will be executed instantaneously. The term event above refers to a change in the state of the model either brought about by some external agency, the input to the model, or as a result of the execution of an activity of some process in the model.

Notice that a process embodies a chain of cause and effect in that one activity can set up a second, which sets up a third, etc. Often such a chain loops back on itself forming a cycle. In fact, such cyclic processes would seem ideally suited to modelling the many and varied cycles of nature, as we shall soon illustrate.

The processes of a model may interact in a number of interesting ways. The most fundamental way is through the medium of the activating conditions. A process operates in the environment created by its own activities as well as those of other processes, and as it tracks through its chain or cycle of activities, its progress may be governed by the prevailing conditions due to other processes.

A second way is that a process can actively intervene to bring to life or to kill another process. A process is brought to life by scheduling one of its activities where previously none was scheduled. Conversely, a process is killed by descheduling its current activity and so making it impossible for an activation to occur. Also, a process may create another process by causing a template for that process (which recall is like a computer program text) to be copied and then brought to life.

The five constructs - scheduling (conditional and unconditional)

of events, bringing-to-life, killing and creating of processes - constitute the primitives of the discrete event modelling formalism. In the same sense, the notion of derivative is a primitive of the differential equation formalism.

### A Generic Example

We shall now discuss a model component called ORGANISM consisting of three processes - LIFE, FOOD-ACQUISITION, and INTERNAL-DEATH. ORGANISM is intended to represent in a sketchy way the life cycle of either an organism or a population of like organisms, depending on the level of resolution desired of the model. By fleshing in the relevant particulars of a set of ORGANISMS and interlinking them with a shared environment, one could arrive at a model of an interacting population (Fig. 2).

The processes are described as follows:

#### LIFE Process

<u>dormant</u>	Hold for time <i>DORMANCY PERIOD</i>
	Wait until <i>CONDITIONS FOR DEVELOPMENT</i> are satisfied
<u>development</u>	Activate FOOD-ACQUISITION process to start from <u>stage?</u>
	Set <i>CHARACTERISTIC</i> <sub>1</sub> = $x_1$
	⋮
	Set <i>CHARACTERISTIC</i> <sub>n</sub> = $x_n$
	Hold for time <i>DEVELOPMENT-TIME</i>
<u>maturity</u>	Activate DEATH process to start from <u>begin</u>
	Set <i>CHARACTERISTIC</i> <sub>1</sub> = $y_1$
	⋮
	Set <i>CHARACTERISTIC</i> <sub>n</sub> = $y_n$
	Wait until <i>CONDITIONS FOR SOCIALIZATION</i> are satisfied
<u>socialization</u>	Set <i>CHARACTERISTIC</i> <sub>1</sub> = $z_1$
	⋮
	Set <i>CHARACTERISTIC</i> <sub>n</sub> = $z_n$
	Wait until <i>CONDITIONS FOR MATING-FERTILIZATION</i> are satisfied



mating-fertilization Hold for time *MATING-FERTILIZATION-TIME*

reproduction Hold for time *REPRODUCTION-PERIOD*

Create copy (or copies) of ORGANISM

Activate its (their) LIFE process(es) to start from dormant

inter-season Hold for time *INTER-SEASON-PERIOD*

Go to socialization

FOOD-ACQUISITION process

stage? If LIFE process is in development go to passive acquisition otherwise go to active acquisition

passive acquisition Wait until *CONDITIONS FOR RECEIPT* are satisfied

receipt Set  $CHARACTERISTIC_1 = u_1$   
 $\vdots$   
Set  $CHARACTERISTIC_n = u_n$

Hold for time *INTER-RECEIPT-PERIOD*

Go to stage?

active acquisition Wait until *CONDITIONS FOR EATING* are satisfied

eating Set  $CHARACTERISTIC_1 = w_1$   
 $\vdots$   
Set  $CHARACTERISTIC_n = w_n$

Hold for time intermeal-period

INTERNAL-DEATH Process

begin Hold for time *LIFE-TIME*

strike Destroy LIFE process

Destroy FOOD-ACQUISITION process

Destroy INTERNAL-DEATH process

In the description, underlined phrases are activity names. *ITALICIZED PHRASES* represent aspects of the description which would have to be particularized to model a particular organism, or species. All other phrases are standard instructions which constitute the activities.

To set ORGANISM in motion the LIFE process would be started in dormant. It would then track through a life cycle which might include bring-to-life the FOOD-ACQUISITION process, the DEATH-process, and the creation of offspring ORGANISMS. The activity names (dormant, development, etc.) represent successive stages of the life cycle. Each stage sets up the succeeding stage by either unconditional or conditional scheduling. Thus the process will remain in the dormant stage for at least a time period *DORMANCY PERIOD* whose actual duration may be a fixed constant or determined by a deterministic or stochastic function of the prevailing values of a certain designated set of environmental variables. After this period, the dormant stage will be maintained until the *CONDITIONS FOR DEVELOPMENT* are satisfied. These conditions may, for example, specify that the values of a designated set of environmental variables be within specified ranges. If, and when, the development stage is activated, the FOOD-ACQUISITION process is activated and the values of specified variables - internal characteristics such as biomass, size, etc. and possibly also environmental ones - are set to specified values. These actions, as well as the scheduling of the next stage, are conceptually instantaneously executed. The reader may now decipher the remaining parts of the model.

#### Discrete Event Modelling of Bio- and Eco-System Interactions

The scheme established in ORGANISM may be fleshed in, and perhaps elaborated or simplified, to model a particular organism or species. For example, a flowering plant beginning as a seed may remain dormant a fixed period or a period determined by the water content of the soil. Germination

(corresponding to socialization) may not occur unless certain conditions of light and temperature are met. Pollination (corresponding to mating-fertilization) may depend on environmental conditions such as the presence of wind and/or on the presence of other organisms such as bees. The latter, of course, would be modelled by specialized versions of ORGANISM and enter the model as model components.

We can begin to see how ecological interactions might arise among ORGANISM components in discrete event model. For example, co-operative linkage would be illustrated by the well known plant bee relation. Pollination activity would be released by the presence of bees. Bee reproduction, on the other hand, would be released by the presence of flowers on the plant to provide the pollen for nest construction.

#### Discrete Event Vs. Differential Equation

Now it may well be that at some lumped level such interactions as above can be modelled, as are classical predator-prey relationships, via net energy transfer rates and differential equations. But it is equally clear that at a more micro level the discrete event philosophy provides by far the more naturally expressive and conceptually stimulating formalism.

One might argue that differential equations can (sometimes at least) be solved analytically while discrete event formalisms appear not to have this feature. In response, we note that realistic differential equation models are usually analytically intractable and require computer simulation. Only in special cases are solutions of equations obtainable after simplifications are made and symmetries enforced. These cases are nonetheless important for the establishment of "natural laws" and the discernment of trends. Given the embryonic state of discrete event modelling, however, it is not all clear that similar developments will not occur.

In any case, there is nothing to prevent the construction of models at various levels employing various modelling formalisms. Interesting questions arise concerning the interrelation and systematization of these models. The answers will be increasingly significant in dealing with complex eco- and bio-systems in the future.

### Modelling Distributed Systems

The rest of this paper deals with the uniform cellular space formalism as an alternative to the partial differential equation (PDE) approach for modelling distributed systems. We illustrate that sometimes it is possible to abandon a traditional formalism and its accompanying world view and yet retain some important mathematical results. At the end of the paper, we briefly describe a computer simulation system which combines the speed and graphic capabilities of computer simulation with the usefulness of a traditional analytical technique.

Many systems that occur in nature are traditionally viewed as very large collections of components distributed in space with interaction occurring between components that are near each other. Each component is usually considered to be essentially the same as every other and is often well understood when behaving in isolation. The usual way of modeling these kinds of systems is through partial differential equations (PDEs). Intuitively, this kind of model might be viewed as a network of an uncountably infinite number of components each interacting with infinitesimally close neighbors. When the components and their interactions are simple enough, many of these equations can be solved analytically using powerful results from functional analysis. However, for most equations one settles for computer generated approximations to specific solutions.

Models within this tradition are ubiquitous in the physical sciences

and have been very successful in accounting for a wide variety of phenomena: wave motion, diffusion, etc. More recently, this modeling approach is being applied to such cases as the intriguing reaction-diffusion systems described by Zhabotinsky and Zaikin (1973) and to population dynamics where the possibility of spatially inhomogeneous populations allows the examination of migratory tendencies or the effects of diffusion on aquatic species. It has also been suggested that processes akin to these might underlie aspects of morphogenetic development.

A less well known way of modelling distributed systems is to view them as cellular automata (also called tessellation automata or homogeneous structures). This approach began with John von Neumann's design for a self-replicating machine. Although he planned to consider a model based on non-linear PDEs, only the conceptually simpler cellular model was completed (von Neumann (1966)). This modelling framework begins with a view of the components as actually separated in space and operating in discrete time. In fact, each component can be viewed as a computer - either a simple one as in Conway's Game of Life which produces a 1 or 0 depending on how many neighbors are emitting 1s or 0s (Gardner (1970)), or a very complex one as in the computing elements of the ILLIAC IV computer.

More generally, a cellular automaton is an array of uniformly interconnected identical finite state automata. Operating sequentially in discrete time, each automaton at each time step receives input from a finite number of neighboring automata and computes a new state for itself. Usually a cellular automaton is described as associated with a finitely generated abelian group - in particular with  $Z^d$  ( $Z$  denotes the integers). In this case  $d$  becomes the dimension of the integral lattice that represents the discretized space occupied by the cellular automaton - a cell, that is, an automaton, is associated with each point in  $Z^d$  as in Figure 3.

Uniformity of interconnection means that one can design a template specifying how a single cell is connected to its neighbors and then determine all the connections by translating this template.

This kind of framework has been elaborated and generalized and has provided a framework for modelling many kinds of systems (see Burks (1970), or the recent survey by Aladyev (1974)). There is increasing interest in cellular automata as models of biological systems that exhibit a degree of spatial homogeneity such as certain areas of brain tissue, sections of heart muscle tissue or colonies of bacteria. Interest has also been generated from pure fascination with the interplay of local and global properties and the remarkable degree of behavioral complexity that can be implied by such simple structural specification.

An important thing to notice is that while one may hope for a closed form solution with the PDE formulation, a cellular formulation is rarely expected to yield a "solution" in this classical sense. Computer simulation is usually the only means possible for gaining insight into a cellular automaton's behavior, and the simulation model is considered not as an approximation but as a realization of the cellular model. In fact, the formalism of a cellular automaton allows us to abandon the idea that modelling with discrete time and space necessarily represents an unfortunate and inaccurate discretization of "real" time and space. It is not necessary to feel that, all else being the same, a finer "mesh" would be better. Although many important things are lost when notions of continuous time and space are abandoned, other things are gained. We feel that in some modelling situations the use of PDEs presents unnecessary mathematical complications. This is especially true since often discrete approximations to the PDEs must be considered anyway. For some applications it may even

be true that the great care taken in numerical approximation introduces an empty kind of precision into the problem since considerable abstraction may have been involved in the initial continuous formulation.

Thus, there are at least two approaches one can take in modelling what we have called distributed systems: the PDE approach and the cellular automaton approach. In some applications it is quite clear which kind of model is more appropriate. When processes appear to be continuous in space and time, the PDE formulation might be more natural along with the accompanying numerical approximation philosophy. When things do not seem to be "really" continuous, a cellular automaton conceptualization might be more valuable. However, in many cases the choice is not clear and is usually decided by the researcher's background or by the tradition prevalent within a discipline. In particular, we feel that PDE's are often chosen because the modeller is unaware of or unfamiliar with other modelling techniques.

On the other hand, discrete formulations often appear to be disconnected from any rich body of lore such as that which exists for PDEs. They seem to be ad hoc constructions that neither benefit from well developed theory nor seem likely to contribute to it. This impression is only partly true. Cellular automata are related to a body of knowledge, but one that is oriented toward questions regarding computational power, formal grammars, and computer design. These sorts of questions have little direct relevance to issues which are important in the context of modelling natural systems. PDEs, on the other hand, can provide extremely useful formulation since enough structure is present to allow useful specification of behavioral questions.

In this paper we discuss systems that can be related to both the PDE and the cellular automaton traditions. It is hoped that in doing this we can combine the conceptual simplicity of a cellular model with some of the behavioral analysis techniques available in the continuous theory. In particular, we will discuss the separation of variables technique as it appears in a discrete situation. While this view is commonly used in the stability analysis of some numerical methods for PDEs, we do not consider the discrete model to be the approximation of any underlying continuous representation. It is thus possible for someone who has no background in the theory of PDEs or in numerical approximation methods to formulate models of distributed systems and to gain an understanding of their behavior comparable to that possible with more traditional approaches.

In order to do this, however, it is necessary to have help from a computer. At the end of this paper we briefly discuss an interactive computer system designed for experimentation with these kinds of systems. Hypotheses can be rapidly tested and the model's structure can be changed easily so that cycles of trials and model refinement can be accomplished relatively painlessly. The models capable of being simulated this way are simple enough so as not to be completely overwhelming but rich enough to provide useful insight into natural phenomena. In addition, we feel that interactive experimentation can help one begin to understand "why" a particular sort of behavior is produced by a particular model.

#### A Subclass of Cellular Automata

The systems we will be discussing can be thought of as special kinds of cellular automata. The state at any time of each of the identical cells can be characterized by a real number or by a finite set of real numbers. In other words, the cell state set is  $R^k$ , where  $R$  denotes the reals and  $k$



is a positive integer (the cells are thus not finite state automata). Each cell receives information from certain neighboring cells which do not have to be immediately adjacent. We make an important restriction on the kind of information about the states of neighboring cells that a cell can use to compute its new state: it must use only a weighted sum of the states of its neighbors. What weight is attached to information from a cell depends on its position relative to the receiving cell. This idea will be made more clear below. Finally, using only this information from its neighbors, a cell computes its new state using any kind of function that can be computed by a computer. Call this function  $F$ . All the cells perform this computation (each using different data) at the same time so that a configuration of cell states changes synchronously to a new configuration.

Figure 4 shows a section of a cellular automaton associated with  $Z$ , i.e. it is one dimensional. The circles represent cells each with state set  $R$ , and the pattern of interconnections shown for cell 0 should be understood to be repeated for every cell in the array. If we let  $q_{n,t}$  indicate the state of cell  $n$  at time  $t$ , then the diagram is intended to mean the following: for each  $n \in Z$ ,  $q_{n,t+1} = F(w_{-1}q_{n-1,t} + w_0q_{n,t} + w_1q_{n+1,t})$ . More generally, we can write  $q_{n,t+1} = F(\sum_{j \in Z} w_j q_{n+j,t})$  noting that a weight  $w_j$  for a connection not shown in the figure can be considered to be 0. Notice that a cell can be considered as a neighbor to itself in cases when  $w_0 \neq 0$ .

We'll continue by giving examples of these systems and describe theoretical results as they apply just to these systems. The general ideas will become apparent without the need for complicated formalisms. The first example is particularly simple in that  $F$  is a linear operator. In such cases the systems are essentially the same as multidimensional linear difference equations that are often used to approximate linear PDEs.

Again, however, we don't need to think of any PDE in order to study their behavior. The first example can be viewed as an attempt to present well known techniques within the cellular automaton framework.

### A Linear Example

This example is a discrete analog of the diffusion process in one dimension. In fact, it is identical to a standard finite difference approximation method for the diffusion equation  $\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2}$  restricted to periodic initial conditions. Figure 5a shows a section of a cellular automaton, but this time is associated with  $Z/N$ , the integers mod  $N$ , instead of  $Z$ . In other words, it is a loop containing  $N$  cells instead of a string of an infinite number of cells. Let the function  $F$  be given just by  $F(s) = s$ , i.e. it does nothing: the new state of a cell is the weighted sum of its neighbors. We indicate this in Figure 5 by not writing  $F$  in the cells.

Under these conditions it is always possible to find an isomorphic system of the simple form shown in Figure 5b. Two automata  $M$  and  $M'$  with state sets  $S$  and  $S'$  and state transition functions  $\delta$  and  $\delta'$  respectively (and no input) are isomorphic when there is a one-one, onto map  $T: S \rightarrow S'$  such that for all  $s \in S$ ,  $T(\delta(s)) = \delta'(T(s))$ . In other words, if  $s_t$  changes to  $s_{t+1}$  in  $M$ , then  $T(s_t)$  changes to  $T(s_{t+1})$  in  $M'$ .

In our case a state  $s$  of cellular automaton  $M$  is a sequence of cell states:  $q_0, q_1, \dots, q_{N-1}$ . The isomorphism  $T$  is the Discrete Fourier Transform (DFT) on  $N$  data points which transforms a sequence of length  $N$  of cell states into another sequence of length  $N$  which we interpret as giving the states of the components of system  $M'$ . It can be seen from Figure 3b that these components change state very simply: the next state of component  $n$  is  $W_n$  times the old state.

A discussion of the DFT is beyond the scope of this paper, but we will state some facts about it. It is analogous to the Fourier series representation of a function and mathematically related to it, but can be understood completely without reference to any continuous theory. Its extensive literature is mostly in the field of digital signal processing (e.g. IEEE (1967) and (1969)), but applications are arising in many areas since a very fast algorithm is available for its computation - the Fast Fourier Transform (FFT). The most important fact for our purposes is that the DFT does not have to be considered as an approximation to anything.

The DFT not only relates the states of the two systems, but also directly relates their structure. Consider the weights  $w_n$  and  $W_n$  in Figure 5 to be values of weighting functions named  $w$  and  $W$  respectively. We should point out here that these are functions of a discrete variable, i.e. they are sequences. We'll talk about them in language usually reserved for functions of a real variable, but they need not be thought of as related in any way to usual continuous functions. In particular they are not discontinuous functions of  $\mathbb{R}$ . With this in mind, if we let  $\bar{w}$  be the function with values  $\bar{w}_n = w_{-n}$  for  $n$  in  $\mathbb{Z}/N$  ( $-n$  is the mod  $N$  inverse of  $n$ ), then  $W$  is the DFT of  $\bar{w}$ . Figure 6 shows how the structure of the two systems are related for specific weights and for  $N = 16$ . Graphs of  $w$  and  $W$  are shown. Remember that they are functions of  $\mathbb{Z}/N$  and not of  $\mathbb{R}$ .

The function  $\bar{w}$  in this example is identical to  $w$  since the latter is symmetric. In general  $\bar{w}$  is different than  $w$  and has the interpretation of a spatial impulse response: if the cellular automaton's initial state is 1 at cell 0 and 0 at every other cell, then after one transition its state will be  $\bar{w}$ . In the theory of PDEs this would be called the Green's function and one would look at the response after some arbitrary time  $\tau$  since nothing

analogous to a single transition exists.

Note that system  $M'$  in Figure 5b is not a cellular automaton since each component can be different (they are all the same only if the cellular automaton  $M$  is already uncoupled). System  $M'$  has  $N$  independent components each of which is a simple one-dimensional linear sequential machine. The behavior of component  $n$  of  $M'$  is given by  $c_n(W_n)^t = c_n e^{(\ln W_n)t}$  for  $t = 0, 1, \dots$  where  $W_n$  is the  $n^{\text{th}}$  Fourier coefficient of  $\bar{w}$  and  $c_n$  is the  $n^{\text{th}}$  Fourier coefficient of the initial configuration of  $M$ . In other words, the components of  $M'$  keep track of the spatial frequencies of configurations of  $M$ .  $M'$  has its "variables separated" or, in automata theoretic terms, is a parallel composition of simple automata.

The system with the impulse response shown in Figure 6 is the same as a stable finite difference scheme for the diffusion equation. Stability is easily checked by looking at the isomorphic system  $M'$ . It is stable because all the weights  $W_n$  in Figure 6 are  $\leq 1$  so that all the spatial frequencies either stay the same ( $W_0=1$ ) or exponentially decay in amplitude. More generally, the  $W_n$ 's will be complex numbers (e.g. when  $w$  is not an even function), and stability results when they all lie within or on the unit circle in the complex plane.

What we have done here works in more general situations but two crucial restrictions remain: the function  $F$  performed by each cell must be linear, and there must be a uniform interconnection of a finite number of cells. This last restriction turns out to mean that the spaces must be looped in every dimension just as our example is looped in one. Alternatively, one can think of configurations of cell states that are infinite in extent but restricted to being periodic along every dimension. This is the view that would correspond to mentally "unrolling" all of the loops. Although

there are other ways of circumventing this periodicity requirement, it often suffices to make the spatial periods large enough so that evolving patterns can be contained within them for the time span of interest.

We again emphasize that no notion of approximation has been used thus far:  $M$  and  $M'$  as described above are exactly isomorphic, their behaviors will never diverge, and we have not ignored anything about integrability, convergence, or roundoff error. Roundoff error re-appears when these systems are actually simulated, but the other kinds of problems arise only when these systems are regarded as approximations to continuous time and/or space systems or as approximations of non-toroidal structures.

Finally, we note that although the example above only deals with a situation in which each cell has state set  $R$ , essentially the same techniques apply when  $R^k$  is the cell state set. In general, cell interaction can be specified by  $k^2$  weighting functions since we need to know how each state component  $i$  of a cell,  $1 \leq i \leq k$ , depends on each state component  $j$ ,  $1 \leq j \leq k$ , of the other cells. The DFT is performed on each of these weighting functions to find the isomorphic system  $M'$ . Each independent component of  $M'$ , however, will have state set  $R^k$ : each one will be a  $k$ -dimensional linear sequential machine or, more traditionally, a realization of a system of  $k$  linear difference equations.\*

Thus far, all the systems discussed have been linear systems since the function  $F$  did very little. The second example is non-linear. We hasten to point out that no general results like those described above are possible in this case. However, by combining the results developed

---

\*The situation is formally the same as that in linear systems theory (e.g. Chen (1970)). Time invariant linear systems can be specified by the impulse response matrix, a matrix of functions of time. Here, the uniformity of a cellular automaton is spatial invariance, and a matrix of functions of space specifies the interconnections. The spatial analog of causality is not required since cells can depend on neighbors all around them.

above with the use of the high-speed algorithm for performing the DFT on a computer, we can describe a useful and rapid simulation technique.

### A Nonlinear Example

This example is based on a model proposed by Dev (1974) and Burt (1974) for the segmentation of visual information into distinct features by the human visual system. Its interesting behavior, however, might usefully model phenomena observed in other research areas. The system is a cellular automaton like that shown in Figure 5a except that the function  $F$  computed by all the cells is nonlinear. Figure 7 shows the shape of  $F$  for arguments between 0 and 1.

One method of analysis, both for systems like this and for similar PDEs is to try to linearize around the equilibrium points and then separate variables. If the weighting function is like the one in Figure 4 (i.e.  $w_0 = 1/2$ ,  $w_1 = w_{-1} = 1/4$ , and the rest = 0), then there are three obvious equilibrium configurations: when every cell is in state 0, every cell is in state 1/2, or every cell is in state 1. These are the fixed points of  $F$  (in the linear case of Example 1 every uniform configuration is an equilibrium configuration as can be seen since  $W_0 = 1$  in Figure 6). Looking at the slope of  $F$  at these points, one would find that the uniform configurations at these values would be stable, unstable, and stable equilibria respectively.

However, in this case, the above kind of analysis misses all of the interesting aspects of the system's behavior. Burt (1974) says that if you think of cells with states near 1 as "on" cells and those with states near 0 as "off" cells, then equilibrium patterns will develop as regions of mostly "on" cells consolidate into completely "on" regions, and mostly "off" regions become completely "off". In other words, stable patterns will be achieved in which regions of "on" and "off" are each larger than some critical size.

Thus, even in a case as slightly nonlinear as this one, the facts described for Example 1 concerning the isomorphism via the DFT seem of little help. With the help of a computer, however, and a very fast algorithm for computing the DFT known as the Fast Fourier Transform (FFT), it is possible to simulate these systems in a way that relates them to the linear case, but not through local linear approximation. In many cases this method is even very much faster than a straightforward simulation.\* Figure 8 is a kind of flowchart illustrating the method. The essential idea is to compute just the linear part of a system's transition function by using DFT. The nonlinear function  $F$  is then computed for each cell in a straightforward manner. This amounts to performing the weighted summation involved in a transition by a high-speed convolution algorithm (see, for example, Stockham (1966)); but, in combination with a visual display at each stage of the computation, it can be interpreted as providing a connection with an underlying linear system.

The numbers 1 - 4 in Figure 8 indicate various stages in the computation at which information can be displayed (we are thinking in terms of useful graphic representation on a CRT). A transition of the cellular automaton proceeds as follows: i) the current configuration (display point 1) is transformed by the FFT into its spatial frequency representation (display point 2). ii) The spatial frequency representation is point-wise multiplied by the function  $W$  (the DFT of the reflection of the weighting function  $w$ ) as indicated in box b of Figure 8. The result can be displayed at point 3. iii) The inverse FFT is applied to find what would be the new configuration if  $F$  were the identity function. This could be displayed at point 4.

---

\* As the networks become larger and more densely interconnected the use of the FFT becomes more advantageous. See Barto (1974) for a more detailed discussion of this point.

iv) Finally, F is applied in a point-wise manner as shown in box b. The result is the cellular automaton's new configuration which is again displayed at point 1.

It is thus possible, at each time step, to see what a related linear system would do in the same situation. The one-step behavior of this related linear system is seen by looking only at display points 1 and 4. The one-step behavior of the isomorphic "variables-separated" version can be seen by looking at points 2 and 3. In other words, the weighted sum can be calculated and observed before F is applied at each cell. If the simulation is observed only at display point 2, one would see the behavior of the non-linear system in the spatial frequency representation. This view will not necessarily be more simple to understand (as it is in the linear case) since the components may depend upon one another. We feel, however, that an intuitive understanding of the DFT together with all four views of each transition can lead to substantial insight into the workings of many cellular automata.

The central idea can be summed up as follows: The method makes use of two complementary state representations so that operations performed in their natural representations (i.e. simple point-wise operations). Part of the problem in understanding a distributed system of the sort described here is that part of its operation will be complex no matter what single representation is used: the spatial representation makes the computation of F by each cell easy to understand but, at the same time, makes the cell interaction difficult to understand; on the other hand, the spatial frequency representation makes cell interaction easy to follow but the operation of F very obscure. The existence of the FFT, however, makes it unnecessary to choose one representation over another.



## An Interactive Simulation System

We have taken this idea of complementary state representations one step further in the design of an interactive computer simulation system. In addition to displaying several representations of a model's behavior, it is possible to allow the structure of the model to be specified and altered by using several representations. All of the weighting functions which, in effect, determine exactly what the interactions among the cells is to be, can be entered into the simulation system by giving either representation. The FFT or its inverse automatically determines the other representation. Referring back to Figure 6, this means that the diffusion example can be specified by entering either the sequence  $w$  or  $W$ .

In our system, implemented to simulate only one-dimensional cellular automata, all functions are specified by literally drawing their graphs with a light pen on a graphic display screen. After a function is drawn, the system is told what role it is to play in a simulation and what representation of it the graph is intended to be. A user can enter a function very quickly in this way. Moreover, by alternately viewing and adjusting the various representations of a function, it is possible to achieve accuracy sufficient for most qualitative experiments.

Since the shape of a weighting function can be so crucial in determining the behavior of a model, and since a function can be changed so quickly and easily, interactive sessions with the simulation system can help build an understanding of "why" a model behaves in a particular way. This is especially true since it is possible to integrate information from two representations of the model's behavior each of which is, in some sense, natural.

## Conclusion

We have presented the "discrete-event" and the "cellular automata" formalisms as alternatives to the more traditional differential equation modelling techniques. The latter have come to be accepted by many bio- and eco-system modellers as being necessary for thinking about the world. It's important to realize, however, that each modelling formalism is natural for only particular classes of real world phenomena - those easily expressible within its world view. Whether or not all phenomena can ultimately be accounted for by a particular kind of model is not at issue here, but rather that certain kinds of models may help us to better understand certain kinds of behavior. We have shown here that the discrete-event and cellular automata formalisms may prove to be fruitful for expressing important bio- and eco-system dynamic relationships.

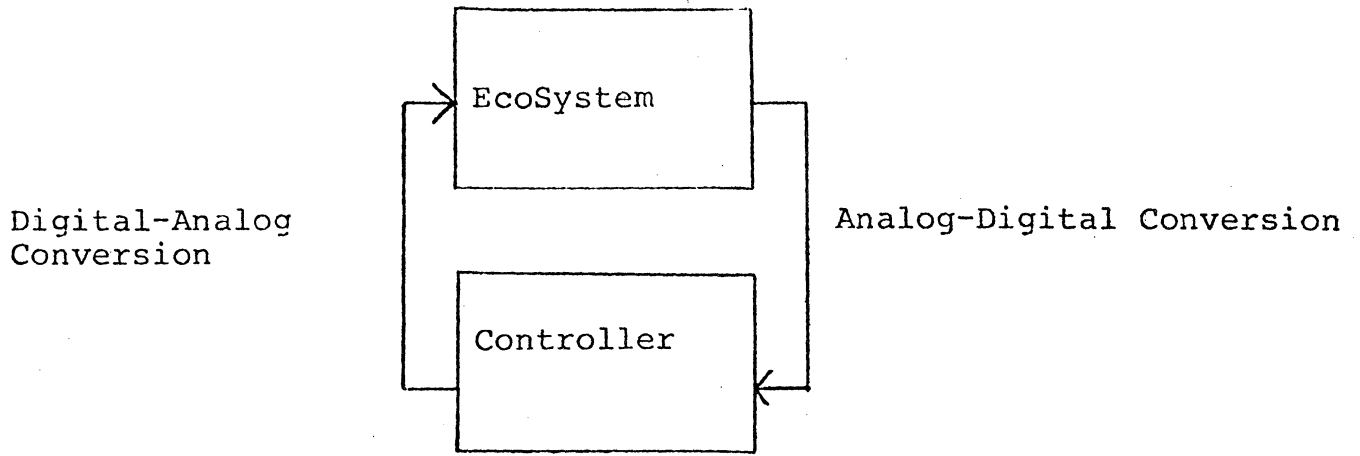


Figure 1

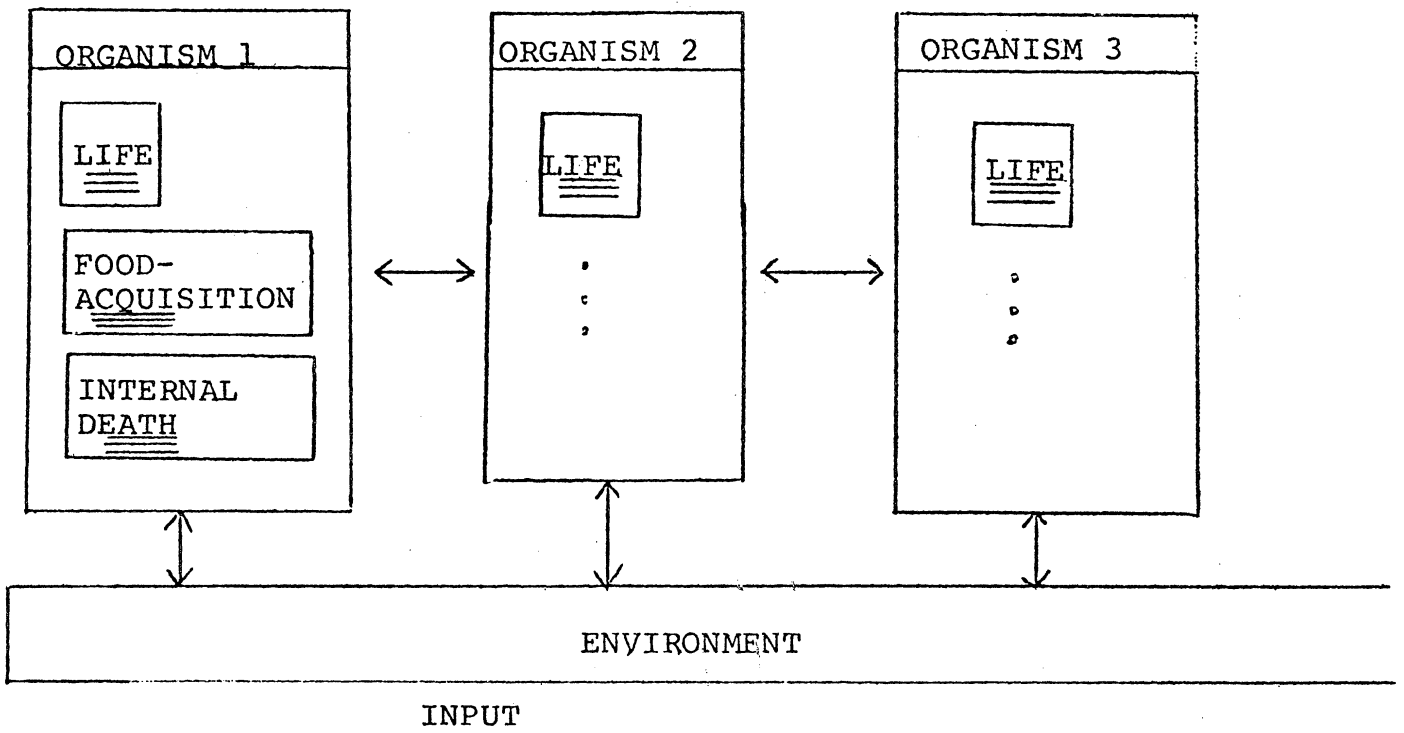


Figure 2

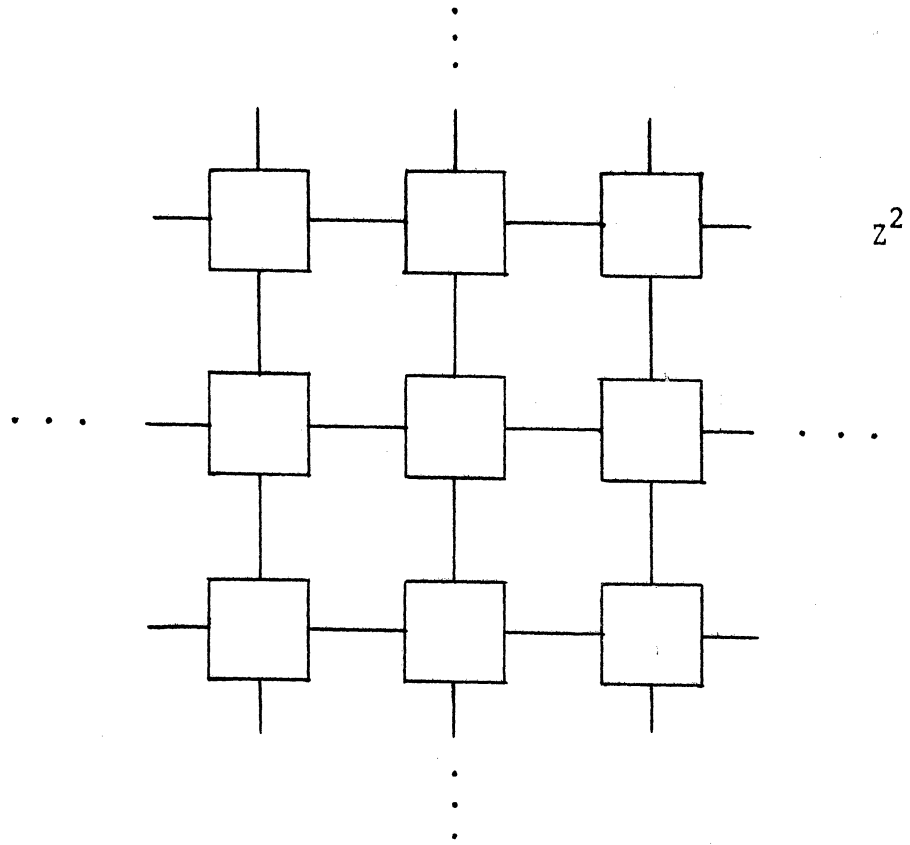


Figure 3

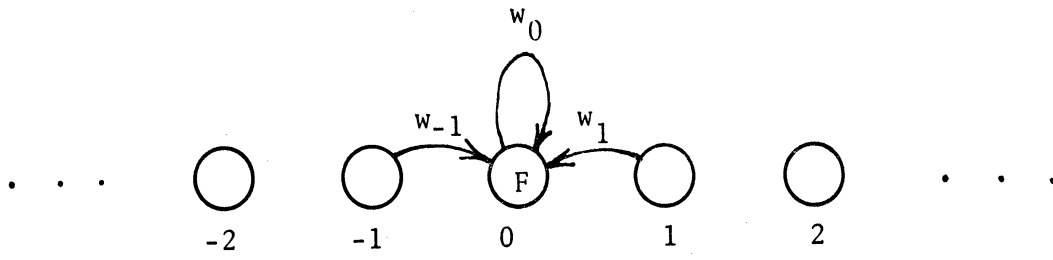
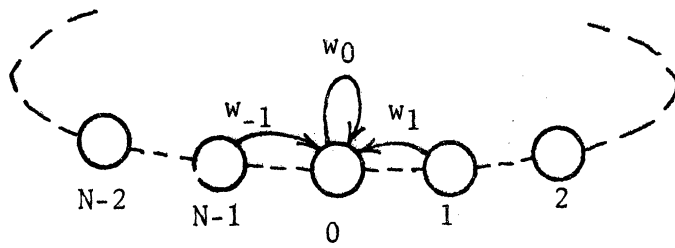
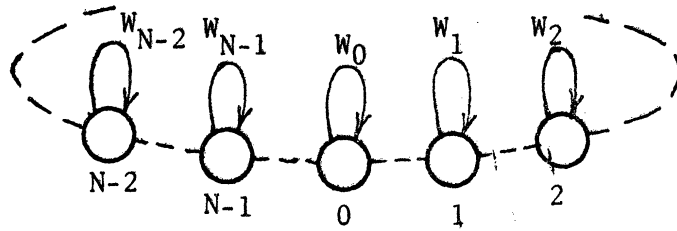


Figure 4



a) Cellular automaton M



b) Isomorphic system M'

Figure 5

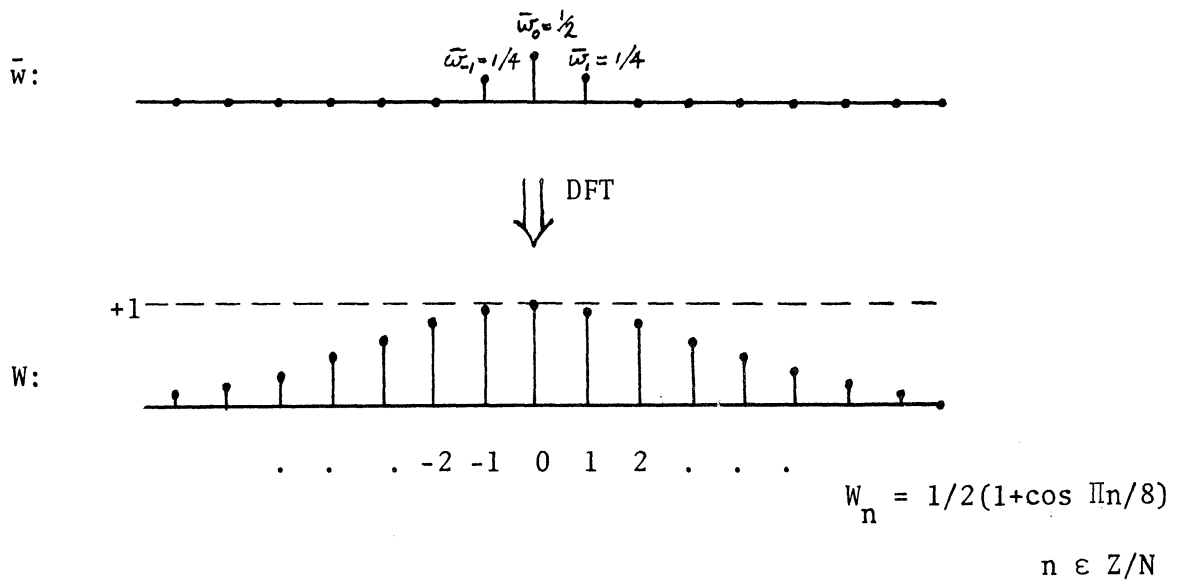


Figure 6

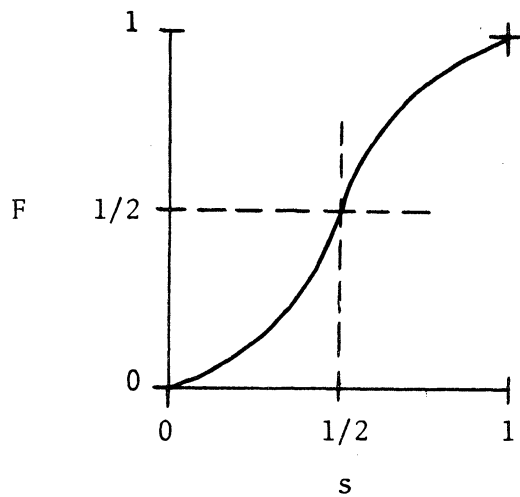


Figure 7

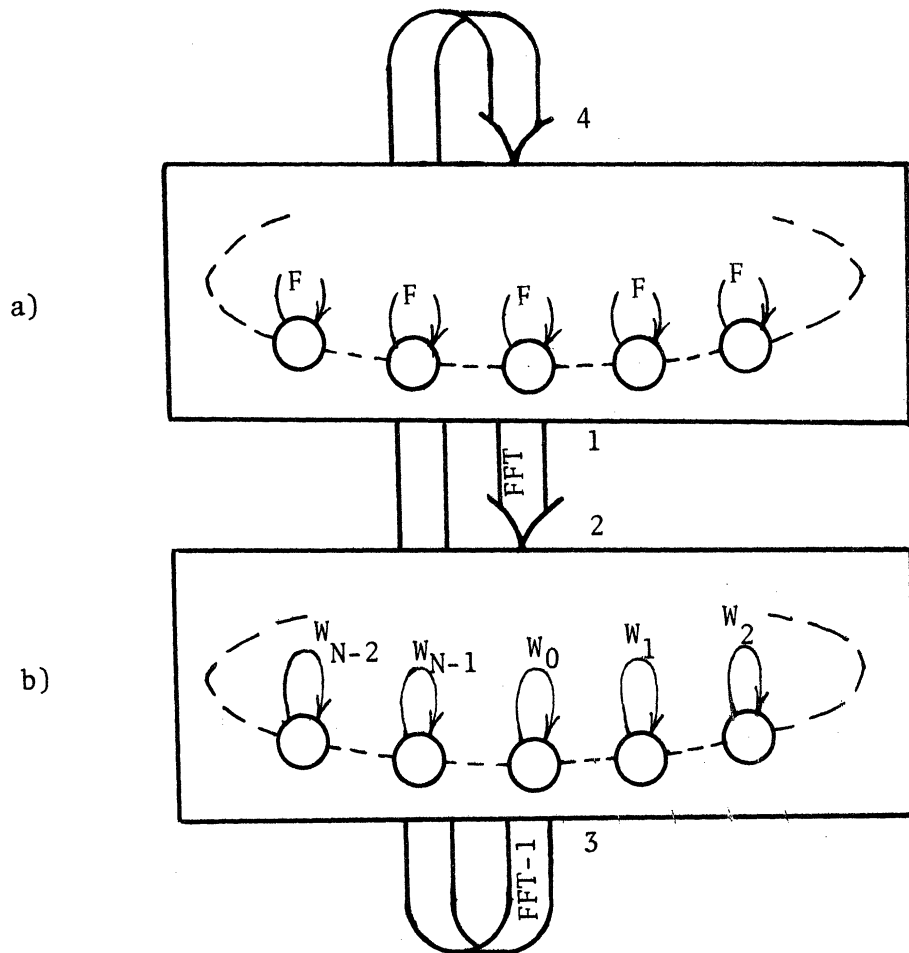


Figure 8



## References

- Aladyev, V., (1974), "Survey of Research in the Theory of Homogeneous Structures and their Applications", Mathematical Biosciences, 22, 121-154.
- Barto, A. G., (1974), "Simulation of Networks Using Multidimensional Fast Fourier Transforms", ACM Simuletter, 5, 4, July.
- Burks, A. W. (Ed.), (1970), Essays on Cellular Automata, University of Illinois Press.
- Burt, P. J., (1974), "Computer Simulations of a Dynamic Visual Perception Model", Proceedings of the 1974 Conference on Biologically Motivated Automata Theory, McLean, Va., June.
- Chen, C. T., (1970), Introduction to Linear System Theory, Holt, Rinehart and Winston, Inc.
- Dev, P., (1974), "Segmentation Processes in Visual Perception: A Cooperative Neural Model", Proceedings of the 1974 Conference on Biologically Motivated Automata Theory, McLean, Va., June.
- Gardner, M., (1970), "The Fantastic Combinations of John Conway's New Solitaire Game 'Life'", Scientific American, October.
- IEEE Trans. Audio Electroacoustics, (1967), Special issue on Fast Fourier Transform and its application to digital filtering and spectral analysis, Au-15, 2, June.
- IEEE Trans. Audio Electroacoustics, (1969), Special issue on Fast Fourier Transform, Au-17, 2, June.
- Sigal, C. E. and Pritsher, A. B., (1974), "SMOOTH: A Combined Continuous-Discrete Network Simulation Language," Simulation, March.
- Stockham, T. G., (1966), "High-speed Convolution and Correlation", 1966 Spring Joint Computer Conf., AFIPS Proc., 28, Washington, D.C., Spartan Books, 229-233.
- Von Neumann, J.; Burks, A. W. (Ed.), (1966), Theory of Self-reproducing Automata, University of Illinois Press.
- Zeigler, B. P., (1975), Theory of Modelling and Simulation, Wiley, N.Y.
- Zhabotinsky, A. M.; Zaikin, A. N., (1973), "Autowave Processes in a Distributed Chemical System", J. Theor. Biol., 60, 45-61.

UNIVERSITY OF MICHIGAN



3 9015 03527 6271