Gaines, B. R. (1987). Uncertainty as a foundation of computational power in neural networks. In IEEE International Conference on Neural Networks Vol.III Caudhill, M. and Butler, C., eds., pp. 51-57. IEEE, San Diego.

UNCERTAINTY AS A FOUNDATION OF COMPUTATIONAL POWER IN NEURAL NETWORKS

Brian R Gaines
Department of Computer Science, University of Calgary
Calgary, Alberta, Canada T2N 1N4

In conventional information systems uncertainty in information is regarded as "noise", and treated as a problem. In neural networks, however, uncertainty is an intrinsic feature of much of the information processing, with many of the computations involving the deliberate injection of stochastic processes. This paper analyzes uncertainty in information representation and processing in terms of the benefits arising, their theoretical foundations, and their practical applications. It gives a new perspective on network computations and the information involved.

1 Introduction

Phenomena in networks such as Boltzmann machines [27] can be treated as those of binary variables subject to random processes. However, there is an alternative perspective which treats the *probability* of a binary variable as itself representing information [4,6,9,10]. Many of the network computations can then be treated as continuous operations on analog variables rather than logical operations on binary variables. A complete range of computational operations in such representations can be carried out in digital networks.

The concept of using the probability of a binary variable to encode an analog value was introduced in 1965 in research on learning machines [18] at Standard Telecommunications Laboratories in England and termed *stochastic computing* [3]. The same approach was discovered independently at the same time at the University of Illinois in research on parallel image processing systems [25]. Research on stochastic computing proceeded at many sites in Europe, the USA and Japan, resulting in two books [22,23] and an International Symposium in Toulouse in 1978 [28]. However, the speed and cost limitations of using conventional digital circuit elements in a random mode restricted practical applications. In 1969 it was suggested that such applications depended on the harnessing of natural stochastic processes such as photon interactions in optical computers [10], and the lack of practical technology has limited developments in stochastic computing in the past eighteen years.

However, advances in technology have made very large scale parallel computing networks feasible, and many of the high-density electronic and photonic devices being developed for such networks exhibit natural quantum-mechanical stochastic effects suitable for stochastic computing. Many proposed networks incorporate random processes although they have not been developed from a stochastic computing perspective. Early work on stochastic computing was targeted on the applications being developed today: machine learning; image processing; Bayesian estimation; partial differential equation solution; pattern recognition and so on. Computational elements, operational systems and some deep theoretical results have been developed for these systems.

This paper reviews stochastic computers, their applications, and theoretical results, suggesting that the stochastic computing perspective of viewing the networks as performing analog computations on probabilities may be valuable in the analysis of a variety of parallel distributed processing systems.

2 Stochastic Computing

The most obvious way to perform stochastic computation in a digital network is to treat the probability that a digital signal will be in the 1 state as an analog variable. If two statistically independent digital signals are brought together in an AND gate then their probabilities multiply. This has been the starting point for the invention, and re-invention, of stochastic computing. Figure 1 shows how this basic idea may be extended to give the additional operations of weighted addition and integration. Addition has to be weighted by the incorporation of an extra random source since the straightforward addition of probabilities could lead to values outside the representable range. A digital counter acts as integrator, and its parallel digital output can be converted back to a random output by comparing it with a random parallel digital source.

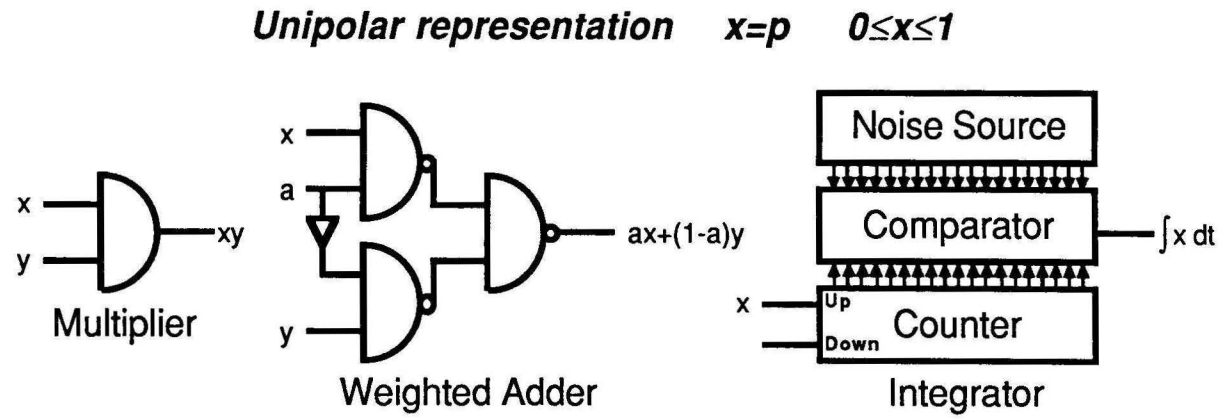

Figure 1 Basic elements of a stochastic computer

2.1 Bipolar Representation

In most applications negative numbers must also be represented, and Figure 2 shows a simple extension which allows stochastic computing to be carried out with variables taking both positive and negative values. An EXCLUSIVE-NOR gate acts as a multiplier in this representation. The weighted adder continues to be effective. A simple invertor gives negation, and hence allows subtraction. Gating at the input of the integrator allows for two summing inputs and a HOLD line.

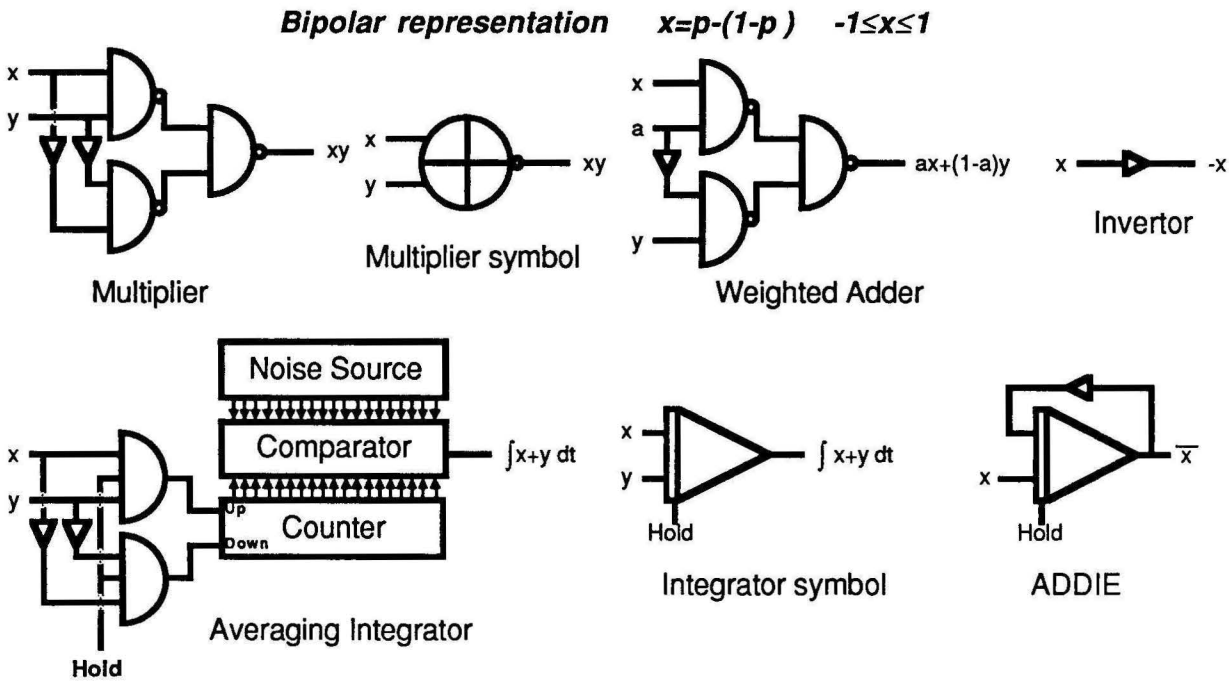

Figure 2 Stochastic computation with positive and negative variables

The noise source and comparator of the integrator act as a natural input device in this representation if the counter is replaced by a parallel digital signal. Analog inputs can be converted by comparing them with a random analog noise source. Negative feedback around the integrator may be used to construct an output interface in that the contents of the counter may be shown to track the probability of the input. The resultant stochastic averaging element shown at the bottom of Figure 2 is an important component in many applications such as machine learning and has been termed an ADDIE (ADaptive DIgital Element) [10].

2.2 Nonlinear Representations

A particularly interesting class of stochastic computations are those in which the representation is nonlinear. For example, the representation p=x/(1+x), where p is the probability of an internal binary event representing an external analog variable x, allows variable in the range zero to infinity to be used in computations. Figure 3 shows that a range of computational operations in this representation can be carried out in digital networks.

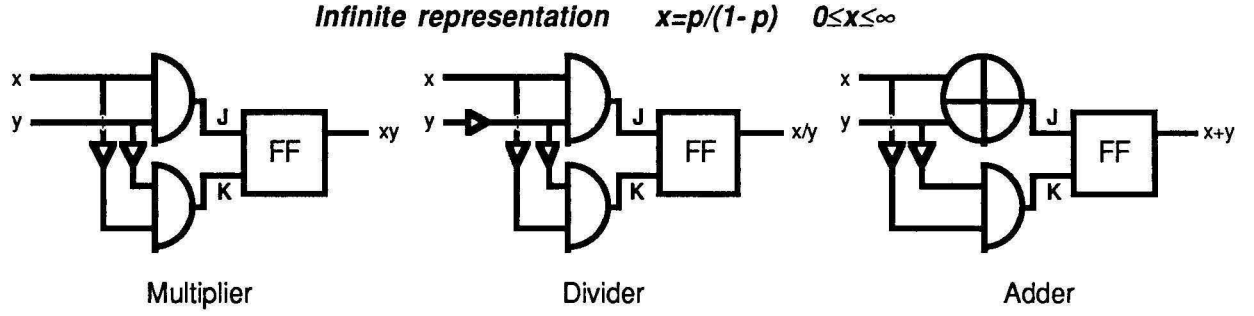

Figure 3 Stochastic computation with infinite variables

An interesting feature of this nonlinear representation is that it shows behavior related to that of the human vision process such as wide range of representation, differential light-dark adaption and Weber-Fechner law of sensitivity. These arise through its logarithmic error sensitivity—that the just noticeable difference in two values is proportional to those values—and the natural input conversion which is a device whose sensitivity decreases with the rate of input and increases by a relaxation process. It may be that the p/(1-p) representation is a natural one for illumination in the neuro-optical system.

2.3 Sigma Pi Units

Many variant stochastic computations may be generated by suitable combinational networks. "Sigma pi" units combining multiplication and addition [27] may readily be designed for each representation as shown in Figure 4. The u and v inputs may be seen as controls varying the effects of the x and y inputs. Such units are known to be capable of generating any monotonic function [27]. However, in stochastic computing it is usually appropriate to design such functions through specific combinational networks rather than by cascading simpler units.

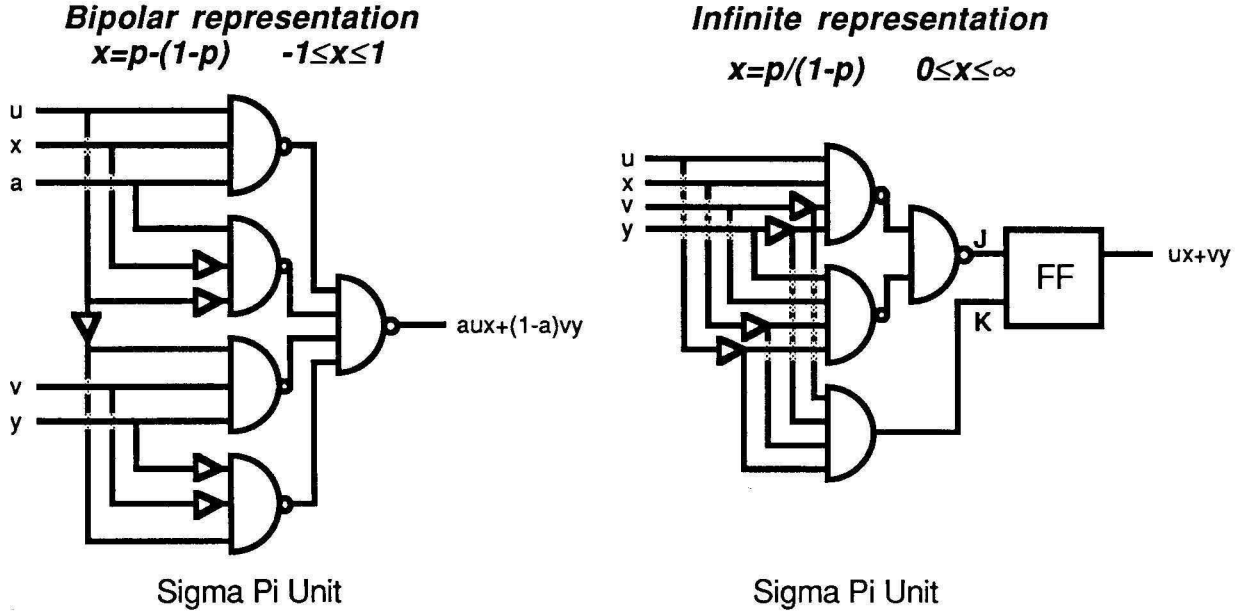

Figure 4 Sigma Pi units with bipolar and infinite variables

2.4 Partial Differential Equation Solution in Networks

Uniform networks of simple stochastic computing elements may be used to solve partial differential equations, such as the diffusion equation, from their boundary conditions. Figure 5 shows a four input summer into an ADDIE used to solve the diffusion equation in two dimensions. Such *thermodynamic* nets have become of great practical interest through the development of the Boltzmann machine [27], and it would be interesting to investigate the relation between digital learning networks and spatial partial differential equation solvers.

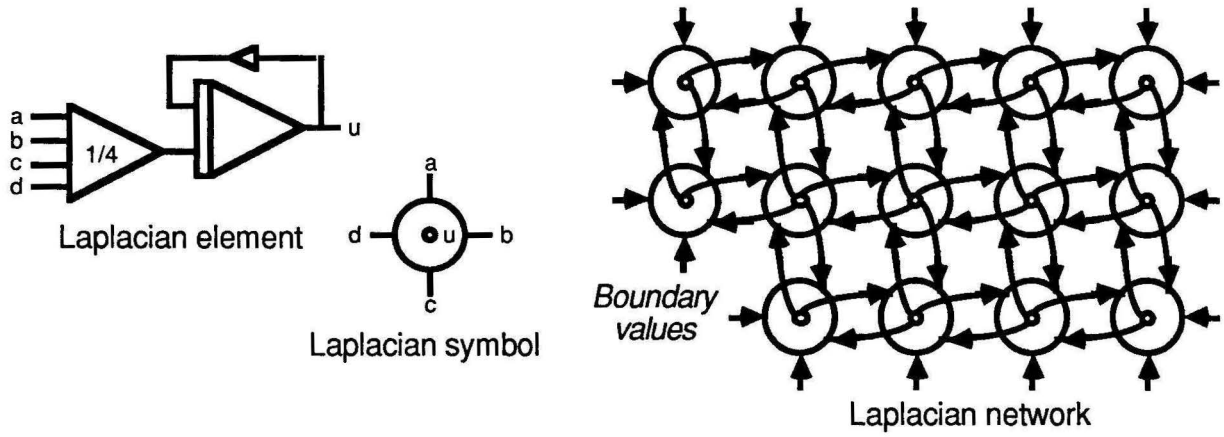

Figure 5 Networks solving Laplace's diffusion equation in stochastic computing

2.5 Stochastic Learning Algorithms

A variety of stochastic computing networks for machine learning have been proposed including those implementing Bayesian prediction [5,10]. Stochastic digital perceptrons have been shown to be more powerful in their learning behavior than deterministic digital perceptrons (see Section 3.2). Practical applications of stochastic learning systems have been studied [22,28].

2.6 Alternative Forms of Uncertainty

The introduction of correlations into the stochastic variables in a network may be used to represent alternative forms of uncertainty. The *phase computer* is a variant stochastic computer in which computations are carried out cyclically with two orthogonal representations, one in which all 1 outputs are collected at the beginning of the cycle (mark-space ratio) and another in which they are maximally dispersed through the cycle (digital differential analyzer) [7,8,19]. This gives very efficient computation in specialist applications, typically solving spatial trajectories. However, it can also be seen as a way of implementing fuzzy logic rather than the stochastic logic of stochastic computing [12,13,17].

2.7 Problems of Stochastic Computing

Stochastic computing has many attractive features. However, there are also problems. The implementation problem has already been mentioned—the use of standard digital circuits in a random mode is inefficient and t he real potential is in the use of naturally stochastic elements. The inefficiency comes from the totally uncoded representation which means that, without randomness, n binary events are needed for a precision of one part in n. The stochastic variance introduced by randomness leads to a square law effect so that n^2 binary events are needed for a precision of one part in n.

The most serious problem is the assumption in stochastic computing that the probabilistic variables are statistically independent. This is easy to achieve in a given layer of the network, but the outputs of a layer are usually heavily correlated with the inputs so great care has to be taken in the design and analysis of multi-layer systems. This is not an overwhelming problem—sometimes the correlations may be used effectively and beneficially—but it reduces the attractive simplicity of the stochastic computing concept.

3 The Power of Randomness

The perspective that sees stochastic phenomena in networks as a form of analog computation on probabilities is useful as a conceptual device. However, the role of truly random behavior in networks also has more fundamental consequences. Stochastic automata are intrinsically more powerful computing devices than deterministic automata, and uncertain networks can perform computations that are beyond deterministic networks.

3.1 Memory Saving

It is known that stochastic automata can define languages that cannot be defined by a deterministic automaton, and that even when an equivalent deterministic automaton exists it will generally have many more states [26]. There are practical examples of this phenomenon in learning applications [30]. For example, consider the problem of regulating a discrete dynamical system to maintain its state within a prescribed region. Represent the system as an automaton, (I,P,S,π,σ), where I is a finite input alphabet, $P\equiv\{0,1\}$ is a binary set of outputs, S is a set of states, $\sigma:S\times I\rightarrow S$ is the next state function, and $\pi:S\rightarrow P$ is the output function. In this formulation is a performance function and the problem is to regulate the inputs to the automaton to cause its output to become, and remain, 1.

Suppose that there is some distinguished element $\lambda\in I$ (the 'zero' input for the autonomous system), and consider the following sets of states:

$$W\equiv\{s: \pi(s)=1\}$$

$$A\equiv\{s: \forall\, n\geq 0, \sigma(s,\lambda^n)\in W\}$$

$$B\equiv\{s: \exists\, n: \sigma(s,\lambda^n)\in A\}$$

W is the subset of S in which it is desired that the state should reside; A is a weak attractor within W; and B is its region of attraction. We assume that both B and S-B are non-empty so that the autonomous system has a region of local asymptotic stability but is not asymptotically stable in the large, and consider the family of control automata whose inputs are from P and whose outputs are in I which induce global stability. For this family to be non-empty it is necessary that B be reachable from S, that is:

$$\forall\, s\in S, u\in I^*, \exists\, \sigma(s,u)\in B$$

where I^* is the free semigroup generated by I.

This problem is one of a class considered independently by Gold in his paper on *Universal Goal-seekers* [21], and by Gaines in a paper on stochastic automata [11]. Both authors demonstrate that no finite state deterministic automaton can act as a universal regulator for this class of problems. On the contrary it is shown that there is a finite state dynamical system which is universally insoluble for all such regulators with less than a given number of states. Both authors give a construction for such a system which has the property that any action by the regulator other than that which it actually does would immediately solve the problem. Gaines calls such a system a *frustration automaton*, and Gold terms the resulting behavior *strongly worst*.

Gold demonstrates that there is a universal primitive-recursive (and hence potentially infinite state) deterministic regulator, and Gaines demonstrates that there is a universal two-state stochastic regulator. This equivalence between recursive and stochastic solutions, together with the non-existence of finite state deterministic solutions, demonstrates the savings in memory and complexity resulting from the introduction of randomness. It suggests a role for random processes in computational and neurological systems in that they enable a simple, finite memory structure to achieve the same capability as a complex structure requiring unbounded memory.

It might be supposed that the stochastic regulator is very much less effective than the recursive one in terms of the time taken to solve the problem. However, a simple example shows that this is not so. Moore's *combination lock* problem involves an automaton whose output is zero until the correct sequence of inputs is given (and the lock opens) [24]. At any stage an incorrect input resets the automaton to its initial state. Moore shows that a sequence of length at least 2^{N-1} is necessary to open an arbitrary binary lock with N states. It can be shown that the mean length of sequence to open the lock with a random generator is 2^N. Theoretical upper bounds on the length of a deterministic solution are very high, but simulation results suggest that the actual mean length is close to that for the stochastic solution [29].

These theoretical results have practical applications. They are the foundations for the use of simple localized random processes to break deadlocks in distributed systems rather than the development of complex central arbitration systems. In terms of Dijkstra's famous *five philosophers* problem [2], one notes that each will get a fair share of spaghetti if, on finding they have only one fork, they toss it in the air and, and if it falls the right side up, put it back on the table for a while so that another philosopher (whose toss has led him to keep his fork) may grab it.

The results also underly the difficulties that people have in coping with randomness in the world [14] and the difficulty of modeling stochastic automata [15,16].

3.2 Convergence of Learning Networks

These theoretical results have practical consequences for neural networks. Convergence theorems for pattern-recognizing networks depend on the continuity of the weights and convergence fails in practice if the weights are discrete [10,20]. Convergence can be re-established both theoretically and practically if the weight changes are made discretely but through biassed random variables. That is, a perceptron-like adaptive network with discrete weights is only able to converge to available solutions if the weight changes are made stochastically.

The phenomenon underlying this result can be seen intuitively if the process is thought of as attempt to follow a steepest descent path in a discrete space. The analog steepest descent point will be approximated by the nearest available discrete point. This does not allow partial descents to accumulate and may lead to limit cycles in which the perceptron repeatedly makes a non-optimal decision.

It is interesting that this behavior was observed empirically in research at IBM on machine learning in the 1960s [1]. It was found that a digital perceptron applied to speech converged to reasonable discriminations but that the same device applied to handwriting did not converge when solutions were available. It was noted that the speech data was 'noisier' than the handwriting data, so noise was added to the handwriting data and the system converged.

4 Conclusions

Uncertainty is a source of computational power in neural networks, and stochastic phenomena are of fundamental importance, not "noise" as in conventional information processing.

4.1 Stochastic Computing

The stochastic computing perspective of regarding digital operations on random sequences as being analog operations on the generating probabilities of those sequences may be useful in the design and analysis of parallel distributed systems.

4.2 Stochastic Representation and Perception

The representation of potentially infinite quantities in the stochastic computer has many properties reminiscent of those of visual perception and may be a useful model of some neuro-optical processes.

4.3 Quantum Mechanical Uncertainty in Electronic and Optical Networks

The natural processes available to implement stochastic networks are those involving quantum mechanical uncertainties, and it may be that the barriers that these are seen as placing on packing densities on conventional integrated circuits should be reconstrued as an opportunity for more powerful forms of computation. Similarly, optical computing elements may offer better opportunities for the implementation of stochastic networks than they do for the emulation of conventional information processing.

4.4 Continuity, Randomness and the Brain

Many of our limitation theorems on conventional information processing derive from assumptions of discreteness in space and time. Continuity in space and time allows for computations that are not subject to Church's and Godel's limitations. However, this is a trivial result unless alternative limitations are considered, notably those of noise. Shannon's channel capacity results apparently map continuous stochastic processes onto discrete deterministic processes in their computational power. However, there are aspects to this mapping that are worth further investigation. Stochastic information may be used in a variety of ways to test hypotheses and there are counter-intuitive results such that hypothesis A may be tested, or B, but not the conjunct of A and B. It is possible that this choice of the way in which to use noisy data is in itself an important aspect of anticipatory processes in the brain. It is suggested that consideration of continuous computational processes may be significant in analyzing the modeling capabilities of the brain and replicating them in networks.

5 Further Reading

The best access to results on stochastic computing is through the survey in [10], the two books [22,23], and the conference proceedings [28].

6 Acknowledgements

Financial assistance for this work has been made available by the National Sciences and Engineering Research Council of Canada. I am grateful to my colleagues Phil Mars, Ted Poppelbaum and Ian Witten for access to their research on stochastic computing and learning systems.

7 References

[1] Clapper, G.L. (1967). Machine looks, listens, learns. **Electronics**, 30th October, 91-102.

[2] Dijkstra, E.W. (1965). Hierarchical ordering of sequential processes. **Acta Informatica 1**, 115-138.

[3] Gaines, B.R. (1965). A stochastic analog computer. **Standard Telecommunication Laboratories Internal Memorandum,** 1-10 (December).

[4] Gaines, B.R. (1967). Stochastic computing. **Spring Joint Computer Conference**, 30, 149-156. Atlantic City: AFIPS.

[5] Gaines, B.R. (1967). Techniques of identification with the stochastic computer. **Proceedings IFAC Symposium on The Problems of Identification in Automatic Control Systems**, 1-10. Prague.

[6] Gaines, B.R. (1967). Stochastic computer thrives on noise. **Electronics**, 72-79 (July).

[7] Gaines, B.R. (1968). Varieties of computer—their applications and inter-relationships. Proceedings of IFAC **Symposium on Pulse Rate and Pulse Number Signals in Automatic Control**, 1-16. Budapest.

[8] Gaines, B.R. (1968). A modular programmed DDA for real-time computation. **Proceedings of IFIP Congress 68**, 1-6. Edinburgh.

[9] Gaines, B.R. (1969). Stochastic computers. Meetham, A.R. & Hudson, R.A., Eds. **Encyclopaedia of Linguistics, Information & Control.** pp. 66-76. London: Pergamon Press.

[10] Gaines, B.R. (1969). Stochastic computing systems. Tou, J., Ed. **Advances in Information Systems Science,** 2, 37-172.

[11] Gaines, B.R. (1971). Memory minimization in control with stochastic automata. **Electronics Letters,** 7(24), 710-711 (November).

[12] Gaines, B.R. (1975). Stochastic and fuzzy logics. **Electronics Letters,** 11(9), 188-189 (May).

[13] Gaines, B.R. & Kohout, L.J. (1976). The logic of automata. **International Journal of General Systems,** 2, 191-208.

[14] Gaines, B.R. (1976). On the complexity of causal models. **IEEE Transactions on Systems, Man & Cybernetics,** SMC-6(1), 56-59 (January).

[15] Gaines, B.R. (1976). Behaviour/structure transformations under uncertainty. **International Journal of Man-Machine Studies,** 8(3), 337-365 (May).

[16] Gaines, B.R. (1977). System identification, approximation and complexity. **International Journal of General Systems,** 3(3), 145-174.

[17] Gaines, B.R. (1978). Fuzzy and probability uncertainty logics. **Information & Control,** 38(2), 154-169 (August).

[18] Gaines, B.R. & Andreae, J.H. (1966). A learning machine in the context of the general control problem. **Proceedings of the 3rd Congress of the International Federation for Automatic Control.** London: Butterworths.

[19] Gaines, B.R. & Joyce, P.L. (1967). Phase Computers. **Proceedings of 5th Congress of International Association for Analog Computation**, 48-57. Lausanne.

[20] Gaines, B.R. & Quarmby, D.J. (1968). The convergence of adaptive pattern classifiers in control systems. Research Memorandum RM 68/2. University of Essex: Department of Electrical Engineering Science.

[21] Gold, E.M. (1971). Universal goal seekers. **Information & Control,** 18, 395-403.

[22] Mars, P. & Poppelbaum, W.J. (1981). **Stochastic and Deterministic Averaging Processors.** London: Peter Peregrinus.

[23] Massen, R. (1977). **Stochastische Rechentechnik.** Munich: Carl Hanser.

[24] Moore, E.F. (1956). Gedanken experiments on sequential machines. Shannon, C.E. & McCarthy, J. **Automata Studies.** Princeton University Press, pp.129-153.

[25] Poppelbaum, W.J. & Afuso, C. (1965). Noise computer. Quaterly Technical Progress Reports April 1965-January 1966. University of Illinois: Department of Computer Science.

[26] Rabin, M.O. (1963). Probabilistic automata. **Information & Control,** 6, 220-245.

[27] Rumelhart, D.E., McClelland, J.L. & PDP Research Group. (1986). **Parallel Distributed Processing.** Cambridge, Massachusetts: MIT Press.

[28] Stochastic Computing. (1978). **Proceedings of the First International Symposium on Stochastic Computing and its Applications.** Toulouse: Institut National Polytechnique (November).

[29] Witten, I.H. (1975). Learning to control sequential and non-sequential environments. Research Memorandum EES-MMS-CON-75. University of Essex: Department of Electrical Engineering Science.

[30] Witten, I.H. & Gaines, B.R. (1976). The role of randomness in system theory. Trappl, R., Klir, G.J. & Ricciardi, L. (Eds) **Progress in Cybernetics and Systems Research III** pp.11-18. Washington: Hemisphere.