SLAC-PUB-4849
January 1989
(A)

Accelerator Diagnosis and Control by Neural Nets*

J. E. SPENCER
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94309

Abstract

Neural Nets(NN) have been described as a solution looking for a problem. In the last conference, Artificial Intelligence(AI) was considered in the accelerator context. While good for local surveillance and control, its use for large complex systems(LCS) was much more restricted. By contrast, NN provide a good metaphor for LCS. It can be argued that they are logically equivalent to multi-loop feedback/forward control of faulty systems and therefore provide an ideal adaptive control system. Thus, where AI may be good for maintaining a 'golden orbit,' NN should be good for obtaining it via a quantitative approach to 'look and adjust' methods like operator tweaking which use pattern recognition to deal with hardware and software limitations, inaccuracies or errors as well as imprecise knowledge or understanding of effects like annealing and hysteresis. Further, insights from NN allow one to define feasibility conditions for LCS in terms of design constraints and tolerances. Hardware and software implications are discussed and several LCS of current interest are compared and contrasted.

1. Introduction

While there are many beam diagnostic schemes for on-line control and off-line simulation, there have been no attempts to simulate a whole facility to predict quantities like integrated luminosity or average, real data rates – presumably because no one can compute the mean probability of failure [1] of LCS. The growing size and complexity has made real-time control so complicated and hard to verify that even 'trivial' changes can make systems unworkable. Since this problem is not limited to accelerators, it is clear that new concepts, hardware and software are needed. In the last conference[2] we explored AI for such reasons. Here we look at NN which are complementary to AI e.g. one finds rules for LCS rather than following models that may be too simple or rigid. Reinterpreted and generalized, ideas from NN provide missing and necessary capabilities for future generation systems.

1.1 Motivations and Disclaimers

Physicists often ignore *noblesse oblige*[3] when other fields are considered[3]. It may be excusable in this case due to the greater evil theory i.e. by an equally pernicious problem in control theory that is widespread and largely ignored. In a sense, the 'field' doesn't exist and so provides no antidotes for ad hoc arguments that arise when old systems are upgraded or new ones proposed. The problem is: How to predict *and* guarantee the performance of LCS? To solve the problem rather than shift it elsewhere, we first try to understand its origins to find means to deal with it.

Finally, a word about terminology. Being interested in general networks of 'things' such as neurons, transistors or people, it is often useful to think of them interchangeably. I hope this causes no problems. Clearly, a transistor is neither a neuron nor any other form of sentient protoplasm with or without taxonomic definition.

1.2 History and Description of NN

Ideas on simulating the brain provide an interesting history. For us, the subject begins with the transistor(1947) – a simple device that its developers compared to the neuron[4]. Their device was much faster and more economical per bit of information. The obvious problem was how to use it in large circuits to best advantage. The simulation of the brain quickly became academic because of the many applications resulting from the transistor's size, reliability and power requirements compared to the vacuum tube. However, it was the early fifties before the idea of the integrated circuit(IC) occurred and 1959 before one appeared.

Even so, it was the mid-sixties before circuit timing in IC's surpassed discrete transistors. By then, the serial as opposed to parallel computing machine was well established due to rapidly improving component switching times and the fact that many people weren't aware of what was or could be available. The situation now seems reversed on both counts. The inherent capabilities of the serial machine[5] are no longer compatible with demand which explains the interest in 'new' materials like GaAs[6]. One reason for the slow progress with such materials is their lack of integration with Si with its broad use and large capital investment. One can view the slow progress with NN or the transistor in related ways. Another reason was stated recently in relation to the development of the transistor: "The field was held back mainly by the reluctance of engineers ... to try, or to learn, something new[7]." While work on new materials is important, it is hard to overestimate the importance of work on new logic elements with more connectivity.

Work by von Neumann in 1952[8] was the stimulus prompting this paper — if not the subsequent development of the field[9]. The acronym 'NN' originated from models of the neuron and its connectivity in the central nervous system. Applicability of such models to other problems such as parallel processing machines and their programming produced the acronym 'PDP' for Parallel Distributed Processing as described in the basic reference[10] on this subject.

1.3 Relation Between AI and NN

Many distinctions have been made between AI and NN such as the ironic one that NN is not a branch of AI. Seymour Papert has described them as two distinct offsprings of cybernetics with AI being the more productive and popular until quite recently. The historical development is an interesting story which explains the clear but controversial distinctions. Many are directly related to the available 'hardware' – the brain and conventional computing machine(CC). Some distinctions which seem valid on technical grounds, at least for now, can be represented as follows:

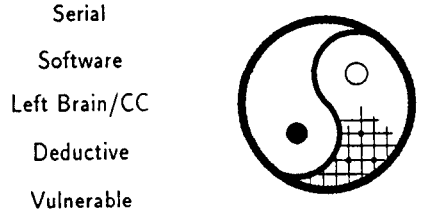

Figure 1: A Schematic Comparison Between AI and NN.

*Partially funded by U.S. Dept. of Energy contract DE-AC03-76SF00515.

2. Reliability, Stability and Durability

LCS rely on and sponsor a 'specialist' system and so risk being dominated by their weakest links without effective feedback and redundancy. Computer and teleconferencing are ways of increasing connectivity in organizations. Grouping experts at consoles in a control room(or boardroom) is another. Such connectivity is useful and must occur but is inefficient and doesn't solve the problem. Real NN don't simply fail or crash when a neuron dies like a CC might. While performance might deteriorate, it could improve given time to adapt[10]. One of many examples follows.

2.1 Importance of Connectivity – The 'Rule of 100'

Neurons fire in $\approx$1 ms and 'decisions' take about 100 steps e.g. response times are $\approx$100 ms[11]. If we cease functioning due to loss of decision capacity e.g. contol of various functions, then for 10^{10} neurons in the brain and a loss rate of $\approx 3\times10^3$ per day(2 per minute) we have a life expectancy of 100 years assuming very high connectivity. Assuming more neurons allows more units of 100 with differing roles and locations. Thus, while we may not use them effectively we definitely don't have more than enough!

3. Prediction and Methodology

We need a method to describe various aspects of LCS that allows us to reduce them to simpler forms while preserving important practical characteristics. Graph theory is especially useful.

3.1 Graph Theory - Undirected Graphs

A connected graph G is composed of sets of links and vertices $\{\mathcal{L},\mathcal{V}\}$ which can be characterized by the number of vertices $\{n_\ell\}$ each having ℓ legs. The total number of links is $2N = \sum \ell \cdot n_\ell$ and the number of loops L is:

$$L(G) = 1 + \frac{1}{2}\sum(\ell-2)n_\ell \ . \qquad (1)$$

Connectivity is defined by the minimum cardinality of either the set of nodes $\kappa(G)$ which separate a graph into subgraphs or set of links $\lambda(G)$ which separate a node from the graph. These integers are a measure of a graph's node and link vulnerability so we may want to make them as large as practicable. However, there are two constraints: cost and complexity which, for simplicity, we take as proportional to the number of links, which we want to minimize. This dichotomy leads to controversies [12] due to inadequate means to assess difficulty and cost consistently.

3.2 Reliability

A particularly useful expression[13] for the reliability of a graph G was given in the context of electronic networks:

$$\mathcal{R}(G) = p_i\mathcal{R}(G*i) + (1-p_i)\mathcal{R}(G-i) \qquad (2)$$

where i is any link and p_i its reliability. $G*i$ and G-i are graphs with i contracted and deleted. Full dots in Fig. 2 are 'terminals' that must communicate and R is the probability of doing so:

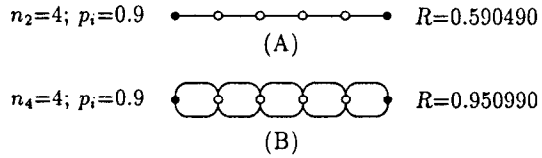

Figure 2: Series and Series-Parallel Separable Graphs.

One can make such graphs arbitrarily reliable but not perfect. Many are impressed with anything that fails once in every 10^6 trys but in LCS this is often inadequate e.g. it's only a second's worth of turns in a storage ring or 1 ms's operation of a high-speed IC element in a mainframe. Fig. 3 shows an example with large increases in connectivity, cost and complexity without reliability.

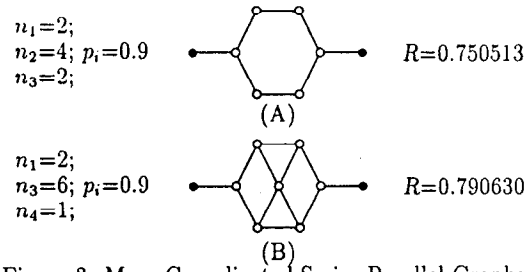

Figure 3: More Complicated Series-Parallel Graphs.

3.3 Simple Cost Analysis

Comparing Fig. 3A to 2A shows a 27% improvement for a 63% increase in cost while 2B gives a 61% improvement for a 100% increase in cost which raises the question of multiplexing.

3.3.1 Single Link Redundancy of Varying Degree

If one or more links are added in parallel with link p_i, increasing the degree of two vertices by j e.g. $\{n_2=4\}\rightarrow\{n_2=2, n_{2+j}=2\}$ in Fig. 2A, there is a fractional increase:

$$\frac{\delta R}{R} = \frac{(1-p_i)}{p_i}\sum_{j=1} p_{i,j}(1-p_{i,j+1}(1-p_{i,j+2}(1-\cdots) \ . \qquad (3)$$

Cost breakeven occurs for:

$$\frac{\delta R}{R} \geq \frac{\delta Q}{Q} = \frac{j}{N} \stackrel{j=1}{\Longrightarrow} p_i \leq \frac{N-1}{N} \qquad (4)$$

where Q is the total cost for the original system with N links and reliability R assuming new links equal in cost and reliability i.e. $p_{i,j} = p_i$. For $N=2$, the breakeven is $p_i \leq \frac{1}{2}$ and for the case shown in Fig. 2 with $N=5$, $p_i \leq 0.8$. We need to go to $N=10$ before $p_i \leq 0.9$ when all of the links cost the same.

3.4 Graph Theory - Directed Graphs

Regular graphs like Fig. 2, with all open vertices of the same degree[14], are undirected until one specifies a sequence of ordered pairs of vertices. We can then speak in terms of arcs or cycles as well as links and let $p_i \rightarrow p_{ij}$ between vertices (i,j). An n-vertex cycle C_n is a path beginning and ending on the same node taking n steps. We call this degree 1 i.e. C_n^1 or simply C_n. For m-traversals or periods we use C_n^m and can talk in terms of an ordered sequence of arcs or periods with sub- or super-periods.

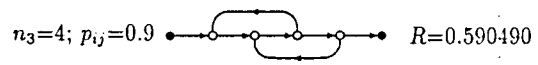

Figure 4: A Simple, Series Directed Feedback Graph

Fig. 4 is reducible to Fig. 2A when antiparallel pairs(annihilation loops) are equal to undirected links with p_i. We also allow p_i to be a function of m or time t so $p_i \rightarrow p_i(m)$. Degree of complexity is taken as the minimum cardinality of vertices between in/out terminals e.g. the shortest directed path.

3.5 Feedback - Stability, Reliability and Memory

Introducing time introduces stability via feedback which improves figure of merit as well as reliability by increasing bandwidth. This increases the equivalent p_i of antiparallel loops but also introduces memory. Manipulating directed graphs can generate feedback loops which generally exist in all systems with at least one dependent and one independent variable. One can associate dependent variables with open nodes and independent with input terminals. In LCS with many variables of both kinds(usually more than we know or can control) one needs to minimize and carefully monitor and control independent variables but this is not necessarily true for couplings between dependent variables. This is one of the more fundamental aspects of the design process and is where NN applies because there are intentional couplings e.g. controllable feedback as well as inherent and unintended.

4. Acyclic Graphs – Neural Nets and Learning

As indicated in Sect. 2.1, neural systems need only about 100 steps for many complex, real-time tasks. People simulate NN on a CC or parallel computer(or coprocessor board) but rarely a 'neurocomputer' such as shown in Fig. 5. It proceeds from input terminals(receptor neurons) to logic units(brain cells) and produces output patterns(on axons/motor neurons) from input patterns(dendrites) and patterns of weights(synapses). Weights and states represent the knowledge base. Weights(or p_i's) occur only on inputs because of potential ambiguities and the directed character of the graph. Feedback, which can make the graph acyclic by autonomously changing weights, is not shown.

With one output per unit a complete propositional calculus[8, 10] is possible. One can think of individual units as multi-emitter transistors or op-amps but a real transistor analog might use ≈5-50K per neuron. With several levels, such systems can set weights i.e. learn and make decisions. A problem is how to determine weights in a changing or noisy environment when little is known.

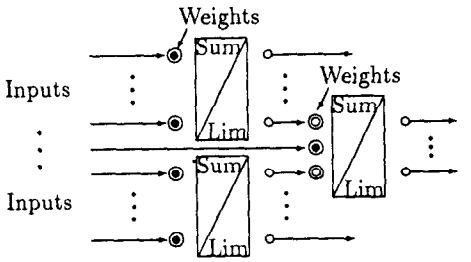

Figure 5: Schematic General Neural Net with Learning

4.1 Examples

One could take a set predicted by experts and vary it[2] or let them vary it to determine an expert system or automaton for a problem. One could diagnose fault modes, avoid random walk, maintain stability or improve 'golden' solutions[2] via real-time feedback where weights transform measured orbit harmonics to magnet excitation harmonics. The training of PDP models[10] on complicated subsystems like klystrons [15] is also possible.

5. Discussion

We were interested in understanding the problems and needs of LCS. Many aspects were considered that can be summarized roughly by Fig. 1. With this split personality, optimal control is virtually impossible. Further, the split runs so deep that it has become a fundamental presupposition in the sense of R.G. Collingwood. NN, as a metaphor for the 'other' helps us find complimentary ways. It provides each of us with an almost ideal, i.e. totally transparent, control system. Ironically, this transparency may explain our oversight of this field. I quote A.N. Whitehead: "Civilization advances by extending the number of important operations we can perform without thinking." Because 'thinking' is defined by the left side of Fig. 1, progress can be defined by the number of problems we are able to pass to the right. Thus, by itself NN is not the answer any more than our current 'idiot savant' system of central CC. However, it provides(or will) a fundamentally different and necessary approach.

By the law of multiplicative probability for serial systems, the more complex a system or unreliable its subsystems, the less the liklihood of success. Control of LCS by large CC's has become a negative example. For accelerators, the storage ring is a positive example because it provides nearly ideal stationary time series based on cycles, natural damping and feedback. Thus, it doesn't need a CC once trained which is all strong justification for SSC and negative for SDI and low rep-rate, conventional colliders.

It seems questionable to attempt higher accelerator energies with old techniques or systems even though they are easier to extrapolate reliably. Old systems provide good benchmarks to gauge new ones *or* to say where and how to upgrade. Scaling the Tevatron, with its present reliability, to SSC is an example. Another is the reliability of SLC klystrons(SLC Handbook) for a collider with 10 times the energy, 10 times the number of klystrons and orders of magnitude smaller spots.

Ultimately, the goal is a method that is computable, consistent and decidable i.e. capable of computation, comparison and optimization. That LCS are generally indeterminant implies only that the design and control are inseparable because feedback, in one form or another, can subvert such effects when fast enough, clean enough and based on an adequate time history. Clearly, one needs new technology that integrates data acquisition, analysis and control in an autonomous way at all levels.

References

[1] There are several weak links e.g. even the idea of testability is often ignored. Scaling from 'known' situations is useful because lattice calculations provide no guarantees even at the level of single particle dynamics. It is also difficult to determine the necessary probabilities in any other way. Considering the various unknowns and sources of uncertainty, cross-talk and noise it seems hard to avoid total grid lock.

[2] T. Higo, H. Shoaee and J. Spencer, Some Applications of AI to the Problems of Accelerator Physics, IEEE Part. Accel. Conf., Vol. 87CH2387-9(1987)707.

[3] E. Schrödinger, *What is Life?*, Cambridge Univ. Press, 1944.

[4] R. Bambrick, Transistor Pioneer Brattain Dies at 85, Electronic News, Vol 33, Oct. 19, 1987. While voltages and currents are roughly comparable, switching times differ drastically. Even so, today's IC transistors, are orders of magnitude slower and dissipate $\approx 10^8$kT more per switch than the theoretical limit.

[5] Vector machines are now available but there are many other possibilities at all levels to be considered.

[6] See cover photo of GaAs transistor with a 1μ feature size that operated up to 30GHz in Physics Today, Vol.23, No.7, Academic Press, NY(1970). GaAs IC's have only recently become available.

[7] Franklin Offner, American Scientist, Vol. 76, No.6(1970)548.

[8] J. von Neumann, Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Comp'ts.,*Automata Studies*, Edited by C.E.Shannon and J. McCarthy, Annals of Math. Studies, No. 34, Princeton Univ. Press, 1956.

[9] J. Anderson and E. Rosenfeld(Eds),*Neurocomputing– Foundations of Research*, MIT Press, Cambridge, MA, 1988.

[10] David E. Rumelhart and James L. McClelland(Eds.), *Parallel Distributed Processing–Explorations in the Microstructure of Cognition. Volume 1: Foundations*, MIT Press, Cambridge, MA, 1986.

[11] J.A. Feldman, Connectionist Models and their Applications, Cog. Sci., Vol. 9(1985)1.

[12] A relevant example involves the Congressional Budget Office and DOE on the cost of the SSC. See Bull. Am. Phys. Soc., Vol. 34(1989)365.

[13] Fred Moskowitz, The Analysis of Redundancy Networks, AIEE Trans on Comm. Elect., Vol. 35(1958)627.

[14] F. Harary, *Graph Theory*, Addison-Wesley, MA, 1969.

[15] M.A. Allen, R.S. Callin, W.R. Fowkes, T.G. Lee and A.E. Vlieks, Reliability and Lifetime Predictions of SLC Klystrons in these proceedings. See also: S.D. Kleban, R.F. Koontz and A.E. Vlieks, IKE: An Interactive Klystron Evaluation Program, IEEE Part. Accel. Conf., Vol. 87CH2387-9(1987)1576.