PROCEEDINGS

DECUS Conference

The papers cited below were presented at the 1970 Spring Meetings of the Digital Equipment Computer Users Society (DECUS) held at Atlantic City, New Jersey. The symposium presentations are reproduced here through the cooperation of Mrs. A. J. Cossette, Executive Director.

The PDP-8/I as a CAT

D. E. SCOTT, Electrical Engineering Department
and
E. DZENDOLET, Psychology Department
University of Massachusetts, Amherst, Massachusetts 01002

A programmed algorithm to compute a weighted ensemble average on a set of output signal realizations is presented. Thus programmed, the PDP-8/I can recover output signals, which are due to periodic input signals, from a noisy system. Arbitrarily, large enhancement of signal-to-noise ratio is attainable. The configuration used is a 4K PDP-8/I with AD08-B analog-to-digital converter.

THE ALGORITHM

Many areas of experimental science are concerned at one time or another with retrieving, from a noisy system, an output signal caused by a periodic waveform which is applied to the input of this system. One typical technique for accomplishing this has been through use of a class of small special-purpose machines called "computers of average transients" (CATs). With the advent of low-cost general-purpose digital computers such as the PDP-8/I with A/D converter, the usefulness of such special-purpose machines has become economically questionable.

In our laboratory, a program has been written for a 4K PDP-8/I with an A/D converter which will retrieve the noise-corrupted periodic response signal from a biological system which is receiving a periodic stimulus signal. This program uses the system's input stimulus as a trigger to reset a coherent averaging algorithm which is applied to the noisy response signal. An arbitrarily large signal-to-noise ratio enhancement may be achieved as a tradeoff against any desired degree of sensitivity to time-varying properties of the response signal. The program stores, as its estimate of the instantaneous value of the sampled response signal, a weighted linear sum of the latest sample and its previous estimate (which is based on all earlier samples of the response signal which were taken at equal delay times after the beginnings of new cycles of the stimulus waveform).

The algorithm starts immediately after the system's stimulus waveform contains a positive-going zero crossing. The system's response signal is then sampled. This sample is divided by a factor, M = 2^n (n = any fixed positive integer less than 10), and temporarily stored; the first entry in an initially empty table of core memory is multiplied by a constant, (M − 1)/M, and added to the temporarily stored new value; then the result is stored back again into the first location in the memory table. After a sampling interval, another sample is taken of the response signal. This sample is weighted and then added as before to the weighted second entry in the memory table. This process continues, each time checking that the stimulus signal has not gone through another positive-going zero crossing, and each time updating the next entry in the memory table. When the stimulus waveform crosses zero with positive slope, the algorithm resets to the first entry in the memory table and continues.

The net effect of the algorithm may be considered to be the creation of an ensemble of sampled response waveform realizations which are all of the same duration as one cycle of the stimulus and are coherent with the stimulus waveform (see Fig. 1). Effectively, then, at each sampling instant a weighted ensemble average is computed. The weighting factors are: 1/M for the sample from the zeroth (most recent realization of the ensemble and (M − 1)/M^2 for the next most recent realization. This is equivalent to a weighting factor of $(M-1)^i/M^{i+1}$ for the ith realization. The ensemble sum of these weighted samples is then computed.

Consider, for example, the case of a noiseless system (one where the response realizations are all identical). The net result of the weighted sum, A_j, is

$$A_j = \frac{1}{M}\sum_{i=0}^{\infty}\left(\frac{M-1}{M}\right)^i R_j \qquad (1)$$

where R_j is the value of the noiseless output at the jth sampling instant. The sum given in Eq. 1 may be obtained in closed form by means of the binomial expansion:

$$(a-bx)^{-1} = \frac{1}{a}\left(1+\frac{bx}{a}+\frac{b^2x^2}{a^2}+\frac{b^3x^3}{a^3}+\cdots\right) \qquad (2)$$

Letting M = a in Eq. 1 and then applying Eq. 2, where b = 1 and x = a − 1, it is seen that,

$$A_j = R_j. \qquad (3)$$

Clearly, then, the correct values, R_j, are recovered by the algorithm in the case of a noiseless system.

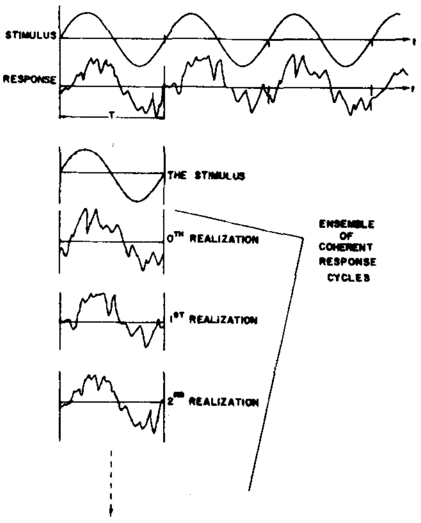

Fig. 1. At each sampling instant the CAT algorithm will perform an ensemble average, automatically assigning the correct weighting factor to each realization of the ensemble.

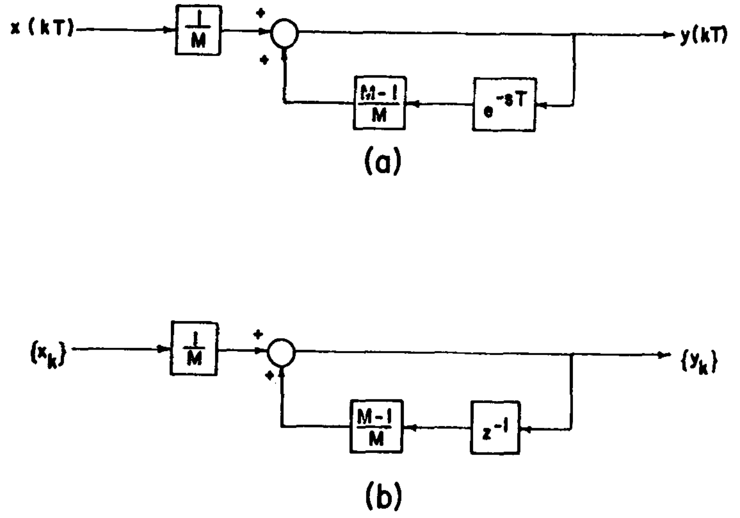

Fig. 2. (a) The CAT algorithm wherein $M = 2^n$ for positive integer values of n. (b) The CAT algorithm in z-transform nomenclature.

If a corrupting noise, uncorrelated with the periodic stimulus and having zero mean value is added to the response signal, it has an arbitrarily small effect on the weighted linear sums, A_j, provided the number of realizations (cycles of output waveform) that are ensemble averaged is large. This is because the expected value of the sum of two varying quantities is equal to the sum of the expected values of each of these quantities, and in this example, as well as in most practical situations, the expected value of the noise is zero.

SIGNAL-TO-NOISE ENHANCEMENT

Rhyne (1969) indicates that the signal-to-noise ratio enhancement obtainable through this method is

$$\left.\frac{\text{Signal}}{\text{Noise}}\right|_{\substack{\text{After}\\\text{Averaging}}} = (2M-1)^{1/2} \times \left.\frac{\text{Signal}}{\text{Noise}}\right|_{\substack{\text{Before}\\\text{Averaging}}} \tag{4}$$

CONVERGENCE TIME

The algorithm considered here is, in reality, a recursive digital filter (see Fig. 2), the input to which is a string of numbers. These numbers are the sampled values of successive realizations of the system's noise-corrupted response signal, each taken at identical sampling times after the onset of a new cycle of the stimulus waveform. Thus, the time interval, T, between these samples is the period of the stimulus waveform.

The net effect of this digital filter is that of a simple first-order RC filter insofar as its response to actual changes in the true (uncorrupted) component of the response signal is concerned. Of particular interest in this regard is the question of the time of convergence of the algorithm. Initially, each location in the memory table is set equal to zero. During the operation of the algorithm, the average value of each of the entries in this table will grow and converge toward a final value. The time constant of this convergence may be obtained as follows.

Let us seek a linear transfer function, G(s), such that the sampled values of its impulse response, g(t), are equal to the series of output values, $[y_k]$, of the digital filter algorithm when the input series $[x_k]$ is the "unit input sequence" given by $x_k = 0$ for $k \neq 0$ and $x_o = 1$. Specifically, we desire a g(t) such that for this "unit input sequence," $[x_k]$, the digital filter output series is

$$[y_k] \triangleq [h_k] = [g(kT)] \tag{5}$$

The z-transform, H(z), of a string of numbers, $[h_k]$, is defined as

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k} \tag{6}$$

The "transfer function" of a digital filter, H(z), is the ratio of the z-transform of its output string divided by the "z-transform" of its input string. The H(z) for the algorithm under consideration is obtained from Fig. 2b as

$$\frac{Y(z)}{X(z)} = H(z) = \frac{a}{1 - bx^{-1}} \tag{7}$$

where $a = 1/M$ and $b = (M-1)/M$.

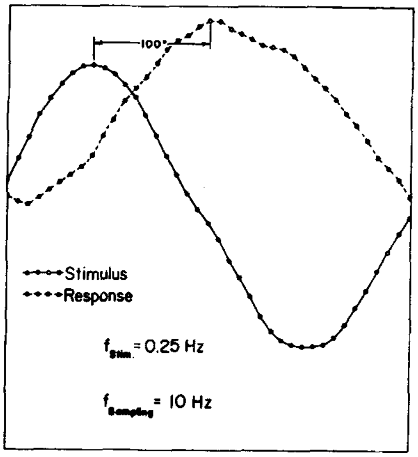

Fig. 3. A selected example of the kind of output typically obtained through use of the algorithm. In this case, the response is electrically induced body sway in a standing human.

From the binomial expansion

$$H(z) = a[1 + bz^{-1} + b^2z^{-2} + \cdots + b^kz^{-k} + \cdots] \tag{8}$$

Hence, using Eq. 6 and then Eq. 5:

$$h_k = ab^k = g(kT). \tag{9}$$

Thus, it is clear that the continuous g(t), in order to have the sampled values given in Eq. 9, must have the form

$$g(t) = ab^{t/T} = ae^{-\alpha t} \tag{10}$$

where

$$e^{-\alpha} = b^{1/T} \tag{11}$$

or

$$\alpha = -\frac{1}{T}\ln b = -\frac{1}{T}\ln\left(\frac{M-1}{M}\right) \tag{12}$$

From the impulse response given in Eq. 10, it follows that the equivalent continuous transfer function is that of the first-order low-pass filter

$$G(s) = \frac{a}{s+\alpha} \tag{13}$$

and so the "time constant" associated with the convergence of the algorithm is given by Eq. 12 as

$$\tau = \frac{1}{\alpha} = \frac{-T}{\ln\left(\frac{M-1}{M}\right)} \tag{14}$$

A TYPICAL EXAMPLE

The algorithm is flexible in that it

will accept a wide range of input frequencies, can supply a wide range of sampling frequencies and will attain a significant enhancement of signal-to-noise ratio. It has been applied quite successfully in our laboratory to determine phase responses of psychophysical systems where noise corruption of the output signal is unavoidable, and where immediate knowledge of this response is needed in order to continue the experiment properly. Only fixed-point, single-precision arithmetic is used, and the program is written in PAL III. Division is accomplished by successive CLL and RAR commands. An example of the output of the algorithm is shown in Fig. 3. The response in this case is body sway in a standing human induced by a low-frequency sinusoidal current stimulus applied to the skull.

REFERENCE

RHYNE, V. T. A comparison of coherent averaging techniques for repetitive biological signals. Medical Research Engineering, August-September 1969.

Simulation of neural sets

BRAD COX*
University of Chicago, Chicago, Illinois 60637

A program has been written for the PDP-8/I computer which simulates the behavior of 64 model neurons which may be connected together to form a functional neural net. The simulation includes provisions for studying the possible mechanisms of learning in real systems. Operation of the net requires inputs from either a robot or a robot simulator program and output from the net serves as input to such a robot, so that the entire system investigates the behavior of an organism and its brain in interaction with an environment.

INTRODUCTION

Mankind has probably been involved in contemplating the paradox of "pondering the ponderer" ever since he first possessed a brain to ponder itself with, but, while there has been some progress in unraveling the functioning of some of the brain's parts, we are still not much closer to understanding the really basic question of how it learns. Part of the problem is due to the fact that neurons are very small, and the parts of the neurons where one expects the most important aspects of their behavior to be carried out (in the synapses and dendrites) are far too small to allow measurements of their activities to be made without damaging them. It is possible to make electrodes small enough to record electrical potential changes from individual cell bodies, but we have little assurance that these are not incidental byproducts of more important events taking place at the molecular level for which we lack the tools to observed directly.

In addition, while technology has been of great help in developing tools to use in experimental brain studies, the concepts that one tends to use when thinking of present-day machinery has little applicability to biological systems. Brains have been likened to hydraulic systems, telephone exchanges, holograms, and computers, but the fact remains that they operate on entirely different principles, for which we lack any good analogs.

There are, however, several characteristics of biological nerve nets which should be of help in understanding how they function. Organisms develop from single cells by following certain rules for development which eventually lead to the creation of sense organs, nervous systems, effector organs, and a body which houses these elements and maintains a fairly constant spatial relationship between them. There is every reason to believe that the brain cannot be studied in isolation from these relationships, and that the feedback loop between effector organs and the sensory apparatus is essential in maintaining reasonable responses to environmental changes and, therefore, normal brain activity.

In addition, as one examines a range of organisms, from those with very simple nervous systems and very stereotyped modes of behavior to those with millions of neurons, one finds that the apparent randomness of connections between individual neurons increases drastically. For example, the sea hare (Aplysia) has a nervous system composed of relatively few neurons which are readily identifiable from individual to individual, with each neuron always connecting to the same other neurons in case after case. In contrast, in mammals, while the gross anatomy is very reproducible, connections between developing neurons appear to be guided by laws that are very general in nature and leave much opportunity for randomness. One tends to suspect that the mechanisms of learning might have evolved to remove the burden of specifying the function of individual neurons from the genetic apparatus, thus allowing larger numbers of neurons, depending on the redundant information content of the feedback loop between muscular activity and sensory changes, to establish their functional identity. The well-known peculiarities of instincts and reflexes probably represent the existence of a backbone of genetically controlled pathways which serve as a bootstrap to enable the organism to survive in his initial contacts with his environment and to keep him functioning until sufficient learned pathways have been established.

This paper describes the current progress of our efforts to develop experimental tools and ways of thinking which we hope will lead to new progress in this field. The neural simulator program is designed to be used as the nervous system for a robot, on which sensors are mounted to encode relationships between the robot and its environment and internal variables. The outputs from the simulator will be used to control servomotors which will alter these relationships. Our next step will be to select some real biological system about which enough experimental information has been published to guide the building of a robot which can reproduce the range of bodily activity relevant to that organism. Ultimately, the entire system will be used to investigate the behavioral effects, modifications of parameters, and even the logical structure of the model neurons described in this paper.

GENERAL SIMULATION LOGIC

The current version of the simulator is operating on a PDP-8/I computer with 8K of memory, extended arithmetic element, and the AXO8 laboratory peripheral unit which is used for oscilloscope control. Up to 64 model neurons can be simulated, each of which can have up to 8 inputs and 8 outputs, which may be connected to any of the other neurons in the net. In addition, interface routines are

*I would like to express my appreciation to Professor Anthony Robertson and to the Committee on Mathematical Biology, for their support and interest, and to the U.S. Public Health Service for providing financial aid under Grant 1 TOI GM 2037.