NEUROPHYSIOLOGICAL DATA ANALYSIS USING A REMOTE CONSOLE COMPUTING SYSTEM

By

Herbert Levitan and Laszlo Betyar

Department of Anatomy and Brain Research Institute
University of California
Los Angeles, California, USA, 90024

Supported by USPHS Grants NB02501-05, NB-05264 (to J. P. Segundo) and 1F2-NB-30,686-01, -02; NASA Grant NsG 505; and Office of Naval Research Nonr 233(91).

FACILITY FORM 602

N68-28836 (ACCESSION NUMBER)

(THRU)

71 (PAGES)

1 (CODE)

CR-95656 (NASA CR OR TMX OR AD NUMBER)

04 (CATEGORY)

Correspond with:

Mr. L. Betyar
Brain Research Institute
Center for the Health Sciences
University of California
Los Angeles, California 90024

ABSTRACT

A time-shared, multiple access computing system with remote consoles has been developed for use by physiologists. This intent is reflected in the design of the input/output console unit, the analog-to-digital conversion routines and the programming language. The capability, versatility and limitations of the system are demonstrated in this communication by describing its use in the analysis of neurophysiological data. The neurophysiological problem concerned the relation between an impulse train providing input to a neuronal junction and the fluctuations in the post-junctional membrane potential thereby produced. The data was therefore of two forms: as a train of impulses and as potential fluctuations in time. Analysis was focused on a statistical description of the interspike intervals and the fluctuations in potential. The mode of analog-to-digital conversion used depended upon the type of analysis to be performed. The capabilities of these routines and their control by way of the console is described. The remote console consists of a 64 button keyboard for input and a memory scope for output. The layout of the keyboard is described and the programming language and technique demonstrated. Some results of the neurophysiological data analysis are shown.

An evaluation of the system reveals that the console's layout and A/D conversion and programming routines are very appropriate and convenient for biologists' use. The almost continuous modernization of the system's hardware and software has enhanced its versatility, but has also caused an inordinate loss of time. This has been the most frequent criticism of the system.

Index Terms: Applications - life sciences; neurophysiology; data processing

Purpose:

During the past several years a time-shared, multiple access computing system of unique design has been developed by the Data Processing Laboratory of the Brain Research Institute at the University of California, Los Angeles. The purpose of this communication is to briefly describe the capability, versatility and limitations of this system with particular reference to the manner in which it was utilized in a neurophysiological research program. Detailed information concerning the computing system (SDS 930 and more recently 9300), the remote console system and its special language, and the results of the neurophysiological study are available else where (Adey, 1966; Rovner, Brown and Kado, 1966; Betyar, 1967; Rovner, Johnston and Betyar, In Prep.; Levitan, Segundo, Moore and Perkel, In Prep.).

The nature of the physiological problem and the form of the data obtained will be discussed first. Next the kinds of analyses desired and the procedures required (such as A/D conversion and programming) for their realization will be enumerated. Finally the console system will be evaluated from the neurophysiologist's point of view.

Neurophysiological Problem:

The research program of which this work is a part is designed to study neuronal integration at the level of the single neuron. Integration is defined here as the relation between the input to and output from a neuron. The input to a nerve cell consists of impulses or spikes in a pre-synaptic nerve fiber (Figure 1.a.). These pre-synaptic impulses set up fluctuations in the membrane potential of the post-synaptic neuron (Figure 1.b.). Whenever the amplitude of the fluctuations exceeds a certain threshold value, inpulses are generated in the post-synaptic cell for propagation elsewhere in the nervous system (Figure 1.c.). In the present study experiments were carried out in the isolated visceral ganglion of the sea slug *Aplysia californica*. Firing was initiated in afferent fibers by direct electrical stimulation with a laboratory stimulator, and the fluctuations in membrane potantial occurring within a neuron of the ganglion were monitored with an intracellularly placed glass microelectrode. Pulses synchronous with the output of the stimulator and potentials monitored with the intracellular electrode were recorded on an FM magnetic-tape recorder. The neurophysiological data was therefore of two types: i) trains of impulses or spikes, of uniform amplitude and duration, with the statistics of the interspike intervals of interest (e.g. interspike interval mean, standard deviation, histogram and autocorrelation functions), and ii) fluctuations in the neuronal membrane potential, with statistics of these fluctuations of interest (membrane potential mean, standard deviation, histogram, and frequency spectrum). Before data processing could be initiated, conversion to digital form was necessary and the appropriate programs had to be written by the investigator in the special console language.

CONVERSION TO DIGITAL FORM:

Two conversion modes are available and their use dictated by the type of analysis to be subsequently performed. One mode converts the amplitude of an analog signal to digital form, the other converts the time interval between specified events in the analog signal to digital form. Both conversion modes can be performed simultaneously.

When the amplitude and fluctuations of a potential are of interest the analog signal is converted into a sequence of numbers whose values and signs correspond to the amplitudes and signs of the potential at the sampled points. The conversion has a precision of 10 bits and sign over a range of ± 1 volt (± 10 volts in the 9300 system), converting the extremes to the numbers ± 128 respectively. Ten times amplification or attenuation of the analog signal is provided in 10 steps by scalars interposed between the input lines and the converter. The digital data is formated into records of 144 or 250 samples in the 930 and 9300 systems respectively. Up to 30 channels of this type of conversion can be performed simultaneously on the 9300 system at maximum sampling rate of 12K samples/second. The computer has a 10 microsecond conversion time which may be fully utilized if transfer of digital values to magnetic tape is not required. On the 930 system 16 channels may be digitized at 3.2K samples/second.

Using the event detection mode of the converter (Rovner, Johnston and Betyar, In Prep.) the time between specified events in the analog signal and their amplitudes are converted into a sequence of couplets. Each couplet is formated into a record as a sequence of 72 complex numbers (125 in 9300 system) where the real part is related to the inter-event interval by the sampling rate and the imaginary part is related to the event amplitude.

When only the interval is of interest, the digital data is formated into records of 144 (or 250 in the 9300 system) real numbers which correspond to the lengths of consecutive intervals. Conversion of this type can at present be performed only on the 930 system on a maximum of five channels simultaneously at a maximum sampling rate of 1.6K samples/second/channel.

The conversion parameters are conveyed to the computer via a conversation mode program called by the user. He specified i) which input channels are to be digitized, ii) the type of conversion desired in each channel, with the option of using one of the inputs as a signal to indicate when to initiate the transfer of digitized values to magnetic tape, and iii) the sampling rate. The sampling rate is the same for all channels; thus when event detection and continuous conversion are performed simultaneously, the sampling rate is dictated by the width of the event to be detected.

When event detection is desired, or when one of the inputs is a signal channel, a threshold must be specified. For the event detection channel, the user may also specify whether the maximum amplitude of a detected event is to be noted and transferred to magnetic tape. When a signal channel is used, he may specify how long to continue the transfer of digitized values to magnetic tape after a signal to begin is recognized.

The user may also process the data in a particular channel while conversion is in progress. Details regarding input channel and the processing program would be specified via the conversation mode, and such conversion and processing could be done on-line, i.e., during a physiological experiment. In the present study all analysis was done from digital tapes, rather than on-line, but there is no difference in the way in which the computer processes on-line and off-line data.

A two buffer technique is used for the transfer of digitized data to tape. As input data is stored in one buffer, a second buffer is transferring its contents to magnetic tape. The output is always faster than the input. When the buffer serving the input is filled, it transfers its contents to magnetic tape while the second buffer serves the input. This cycle is repetitive and the time allotted to on-line analysis is limited to that of one complete input/output cycle. When the processing time exceeds the cycle period the system displays a warning on the user's console scope. The user must then utilize a less ambitious program for on-line analysis or reduce the sampling rate.

To initiate conversion to digital form the <u>RUN</u> button on the user's console is pressed. Transfer of digitized values to magnetic tape begins immediately if no signal channel is used, or awaits the signal if one is specified.

As the digitized data is transferred to magnetic tape in blocks of 144 (or 250) numbers, each block is labeled. The labeling consists of a letter A through Z and four special characters referred to as ";", "<", "=", or "≥" (which correspond to input channels 1 through 30 respectively), and two numbers. One of the numbers, the "continuation number", indicates the order of the block in a sequence of such blocks; the other, denoted "data level" (k), distinguishes between conversion runs. For example if the first data string transferred to digital tape was derived from the second input channel and consisted of 280 digital samples, it would be broken into two blocks labeled B0kl and Blkl. The letter B refers to the second input channel, continuation number "0" contains the first block of 144 (or 250) samples with the remaining 136 (or 30) filling the first part of the next block, and "kl" specifies the first conversion run. The digitized data is referred to by these labels for subsequent display and processing. An example of some calibration signals converted by the event detection and analog mode are shown in Figure 2a and b, respectively.

PROGRAMMING:

Once the investigator has specified the type of analysis he requires and has selected the appropriate conversion mode, he must construct the processing programs and present the results in a useful form. In this section the remote console's layout will be described and the method of programming demonstrated.

The remote console is an input/output device that allows direct intercommunication with the computer. It consists of a 64 button keyboard for input and a memory scope for output. Each button may be interpreted in an upper and lower case, where the upper case is an operator and the lower case an operand. Some of the operator buttons act directly on the contents of the accumulator, others require that an operand (number or address) be specified. Although the lower case meaning of each button is constant, the upper case meaning may be varied. Each set of 64 operators is called an operator level and 60 such sets are possible. Ten of these, donated system levels, contain basic arithmetic and systems programs coded in machine language, and are available to all users but cannot be altered by them. The remaining 50 levels, denoted user levels, are identified with alphabetic letters and symbols. The meaning of almost all the buttons on these levels may be altered by substituting a user's program. Thus, each of these levels may include a combination of systems programs and user programs. The user levels are the private libraries of each user. Unless altered by the user each button on the user levels has the definition listed in Appendix A.

A user may generate a program by stringing together system level programs and/or previously defined user's programs. He must first define the user level desired, select and press the button under which the new program will be placed and choose and press a four letter name for the program. After that, the program is specified by pressing the desired keys in order. The length of any program is limited to 114 (or 250) button pushes, but by referring to other user programs as subroutines there is practically no limitation to program length. The interpretation and execution of a user program is done by the console programming language which is called the Shared-Laboratory-Interpretive-Processor (SLIP).

To illustrate how a short program would be written consider the problem of generating the function (using Fortran notation)

(2*PI)**(-(.5))*EXP((-(.5))*X**2)

where X represents the contents of the accumulator and may be a single number (scalar) or a string of several numbers (vector). We choose to replace the upper case function of the button WAIT on user level A with the program named GAUS through the following sequence of button presses: LEV A° PROG WAIT G A U S°↑ 2° /-2° EXP /2.5056° RETN END; where (2*PI)**(.5) = 2.5056, and each underlined word is one button press, as are the other characters (see Appendix A). Thereafter, when the WAIT button is depressed when working on user level A, the Gaussian function of the accumulator contents (X) is obtained and the result left in the accumulator in place of X. This program can also be used as a subroutine and referred to in subsequent user programs.

The development of more complex programs, such as those used in the present study, requires similar but more lengthy coding. The first consideration is the type of description desired. Next a flow diagram is constructed which shows how the blocks of data on digital tape should be retrieved and manipulated to give the desired display of results, and what initialization parameters are required. The blocks of the flow diagram are then rewritten in the console language. Finally, the total program is divided into subroutines based upon the imposed limitations on a single program's length or the function of a block in the flow diagram.

Debugging of the program begins with the assumption of some dummy data and a stepwise examination of the program on paper. After obvious flaws are corrected by this routine, the program is entered into the computer by the method illustrated above and tested again, either with computer generated data or with real digitized data whose parameters are already known. Under ideal conditions, i.e., in the absence of any computer generated problems, conversion to a working program is very rapid, due to the rapid turnaround time.

ANALYSIS OF DATA:

As noted above, the neurophysiological data in this study was in the form of fluctuations in membrane potential and trains of nerve impulses. Upon conversion the digital data consisted of blocks of numbers related to the sampled amplitudes of the potentials and the intervals between consecutive impulses.

Two main programs were developed to analyze interspike interval statistics. One determined the interspike interval mean and standard deviation of the train and displayed the first order interval histogram (Moore, et al., 1966); a listing of this program is given in Appendix B. The second major program displayed the autocorrelation function of the impulse train (Perkel, et al., 1967).

One of the objects of the present study was to compare the neuronal response to random and regular input. To provide random input, a Geiger counter exposed to a weak radioactive source was used to trigger stimuli to afferents of the ganglion. Except for the fact that the interval between two events can be no smaller than the deadtime of the counter, such a stimulus train resembles a Poisson process (Feller, 1957). The interval histogram should be nearly exponential and the autocorrelogram nearly flat. To obtain the interval histogram the digital data was processed one record at a time. As each record was processed its mean interval and the cumulative interval histogram were displayed on the scope. At the conclusion of the process a cumulative interspike interval mean, standard deviation and first order histogram were displayed. Processing of one such random train containing 1440 intervals took 33 seconds for 144 samples/block or 25 seconds for 250 samples/block and is illustrated in Figure 3.a. The autocorrelogram program operates on data segments of length equal to the specified range of the autocorrelogram and the results of successive segments are summed. This program took an average of 500 or 1000 seconds to process 1440 intervals in data blocks of 250 and 144 samples respectively, and displayed the results shown in Figure 3.b.

To compare the effect which inputs of varying statistical character have upon the output of a neuron, histograms of the resulting fluctuations in membrance potential were obtained. Such a histogram estimates the probability density function of the membrane potential. It is a display of the relative time spent at a particular membrane potential on the ordinate against the membrane potential on the abcissa. The area under the histogram between any two values of potential estimates the fraction of time during which the potential lies between these values, and the area to the right of a particular potential gives the fraction of time the potential is above that value. The potential histogram of a sine and triangular function are illustrated in Figure 4. These were used for checking the correct execution of the program. The program developed for determining the membrane potential mean, standard deviation and histogram is also shown in Appendix B. Except for the initialization of such parameters as the data label, and the range and bin width of the displayed histogram, this program utilizes the same subroutines as the interval program to calculate the mean and standard deviation, and display the histogram. It thus takes about the same time to process 1440 samples of potential data.

Figure 5 compares the potential fluctuations under regular and Geiger-driven stimulations at a mean rate of 3/second. Each input form produces a different membrane potential histogram, as seen at the bottom of the illustration. A narrow relatively uniform histogram was derived from regular excitation, while a broad skewed histogram was produced by Geiger excitation. The curves have been normalized so that the area under each curve is the same. It is evident from these histograms that the two inputs will in general not be equally effective in evoking discharge from the cells. For a spike threshold, like A in Figure 5, that is relatively high, i.e., less negative than the mean depolarization level, the Geiger-driven stimulus would be more effective, since the corresponding histogram indicates that the potential spends a greater fraction of time above threshold. Contrastingly, with a threshold like B, more negative than the mean depolarization level, regular stimulation would be more effective.

The neurophysiological study has thus far been limited to the examination of the relationship between pre-synaptic spike train statistics and the membrane potential histogram of the post-synaptic neuron. Spectral analysis of the potential fluctuations in the post-synaptic cell is anticipated in the near future but user programs have not yet been developed. Through the use of such systems programs as *ZERO* and * (see Appendix A), which are already available, and provide an indication of zero crossings and the convolution of the two functions, respectively, a user program for spectral analysis should be readily forthcoming.

EVALUATION:

In the preface to a monograph analyzing time-shared computing systems (Scheer, 1967), Teager outlines four interconnected factors upon which the performance of a computing system depends: i) "the application and user population," ii) "the language and device interface through which communication to the machine is achieved," iii) "the internal programming systems that translate user specifications into machine processes," and iv) "the central processing and accessible storage capability of the equipment *per se*." He adds the obvious, though important comment, "the observed behavior of a system is largely dictated by the weakest link in the user-language-software-hardware chain." We will evaluate the present system in these terms.

The user population has consisted mainly of biological scientists whose primary interest is in neurophysiology. Those more comfortable with mathematics tend to seek the computers aid more frequently. Prior to their exposure to the present system most had minimal computer experience and that experience was usually via an intermediary. Potential users have approached the computer with the intent of utilizing it for a spectrum of tasks such as: "desk-calculator" computation, data reduction, stimulation or modeling and on-line experimentation.

The programming language and device interface are quite appropriate for the intended user population and their requirements. The "words" of the language are mostly self-explanatory but an introductory course and manual are indispensible aids for gaining an appreciation of the word meanings and programming techniques. Most rapid learning occurs at the console itself through experience. The rapid turnaround time afforded by the console system is particularly helpful at this stage for reinforcement of clever usage or dramatic indication of errors. The rapid turnaround is also enormously helpful to more proficient users for the development and debugging of new programs.

The language and programming procedure are advantageous from another point of view. Experience with this type of device leads to an appreciation of the capabilities and limitations of computers in general. It impresses upon the investigator the importance of routinely and carefully checking even established programs with a calibration signal of the same type and range as his actual data. The exposure also teaches the investigator how to phrase the questions he wishes to ask of his data so as to quickly arrive at a computer aided solution.

The software or systems programs offer a great variety of basic operations and have been quite adequate for handling the kinds of problems dealt with thus far. The conversation mode for controlling analog-to-digital conversion is easy to use and quite versatile. Its prime limitations are i) the low real-time sampling rate available for event detection (1.6K samples/second) and ii) the restriction of a common sampling rate for all input channels regardless of conversion mode. Both these limitations will be rectified in the near future.

The fact that the system is still in the development stage has posed some serious problems for users. When the 930 system was working relatively reliably it was replaced by the larger, more versatile 9300 system. The new system offered the advantages of increased speed and memory size, which in turn meant faster conversion rate and execution time, more system programs (e.g., convolute operation) and the capability for indirect addressing. A major innovation was the addition of a disk storage device. This permitted very rapid storage and retrieval by several users without the slightest indication that other users were present. Previously a magnetic tape unit was used as the common scratch or working area, and retrieval and storage delays varied from "imperceptable" (300 msec.) to "unbearable" when two users tried to access the same tape (Walter, 1966).

The advantages were balanced however by the introduction of problems almost inherent in the installation of a new system. The delays caused by system errors and the debugging of converted system programs limited the time competent users were willing to spend utilizing established programs or developing new ones, and some potential users avoided the computer entirely. The point frequently was reached where the user was not sure whether the trouble lay with himself or with the machine.

In summary, the language, device interface and software are quite appropriate and convenient for biologist's use. The most frequent criticism has been aroused by delays due to breakdowns in the central processing system. Since these are at their maximum following the system's installation and are often significant even after a year, it is felt the frequency of such major changes be minimized and that existing systems be utilized until work is seriously handicapped by the inherent limitations.

APPENDIX A

INTERPRETATION OF CONSOLE KEYS IN THE BASIC SYSTEM LEVEL (1).

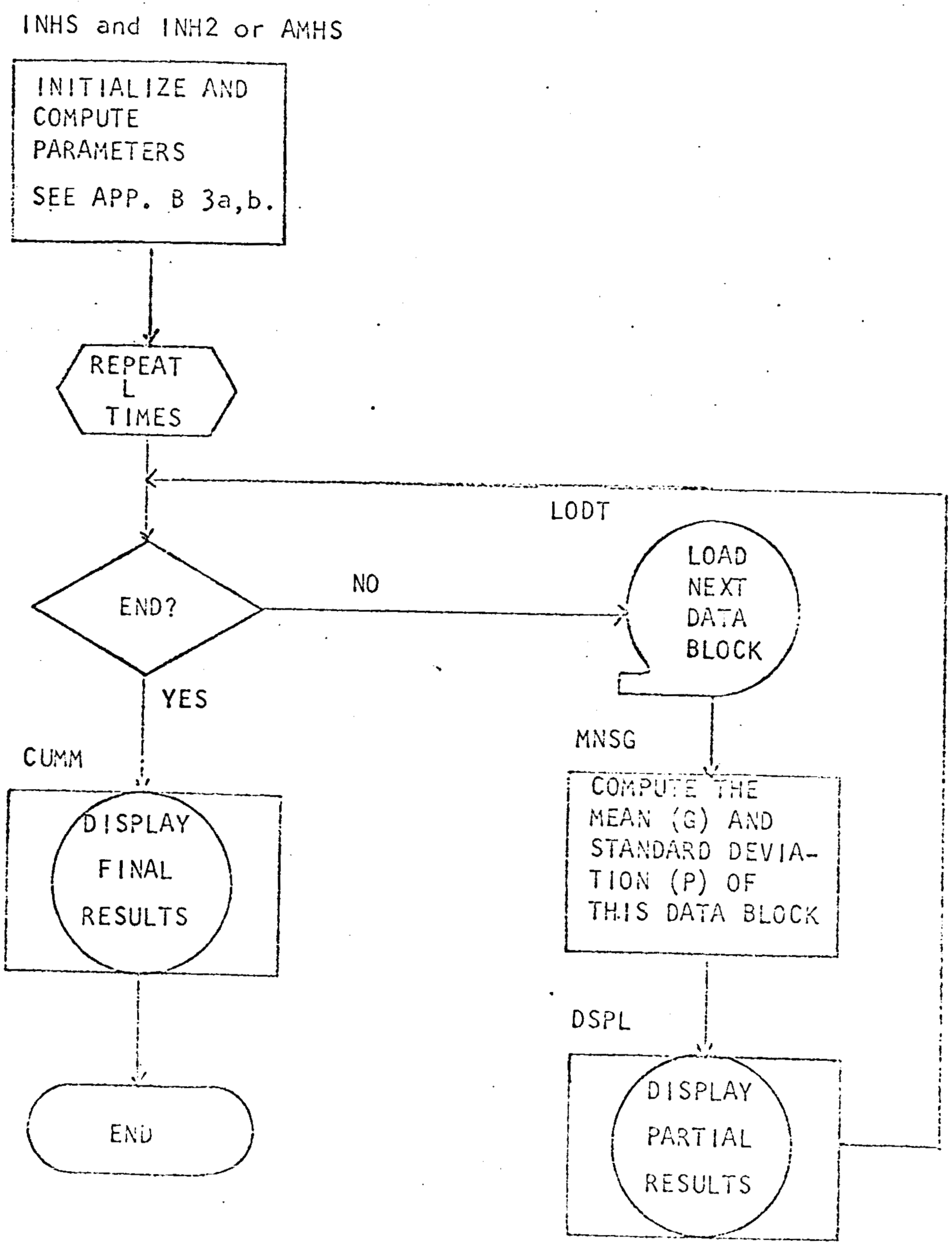
UC	LC	KEY	FUNCTION OF THE OPERATOR
RETN	0	00	Program terminator
WAIT	1	01	Wait requested
CONT	2	02	Continue operation
BUG	3	03	Debug routine
LIST	4	04	Alphanumerical display of AC
TYPE	5	05	Print on scope
RUN	6	06	Starts or stops a previously defined background activity
SET	7	07	Parameter set-up
BLNK	8	10	Scope erase
PLOT	9	11	Vector display of AC on scope
FIND	Q	12	Search a value in the AC
EXT	W	13	Extract element(s) from AC
PRGL	E	14	Load program to AC for correction
PRG→	R	15	Store program from AC
TRNC	T	16	Truncate the values in AC
HEDL	Y	17	Examine the header of AC
TMOD	U	20	Modify the type of data in AC
MODE	I	21	Select scope at plot mode
CURS	O	22	Extract and extend a part of AC
INC	P	23	Increment or decrement continuation index or level
SHFT	A	24	Change the initial index of AC
ROT	S	25	Rotate AC to the left or right
FLIP	D	26	Invert the order of values in AC
MOD	F	27	Execute module n
MIN	G	30	Find the smallest element of AC
RAND	H	31	Generate a random vector in AC
ZERO	J	32	Count zero-crossings in AC
INTP	K	33	Linear interpolater
AVG	L	34	Compute the average of AC
SGMA	;	35	Compute the standard deviation of AC
SIN	Z	36	Compute the sine of AC
COS	X	37	Compute the cosine of AC
ATAN	C	40	Compute the arctangent of AC

UC	LC	KEY	FUNCTION OF THE OPERATOR
LOG	V	41	Compute the logarithm of AC
EXP	B	42	Each element of AC acts as an exponent of e
ABS	N	43	Replace AC with absolute values
HIST	M	44	Distribute values in AC with present bin width
CONJ	,	45	Set the signs of values in AC
X	.	46	Multiply
/	/	47	Divide
<	<	50	Logical operator less than
=	=	51	Logical operator equal
≥	≥	52	Logical operator greater than or equal
SUM	(	53	Sum the AC
Δ	Δ	54	Compute the delta between successive elements of AC
PROD	)	55	Compute the product of elements of AC
*	*	56	Filter the AC
↑	↑	57	Raise AC to an exponential
+	+	60	Add
-	-	61	Subtract
SPACE	SPACE	62	No operator
LOAD	CR	63	Load data into AC
PROG	↓	64	Signify subsequent key-presses as a user generated program
END	←	65	Program or repeat loop end indicator
DATA	o	66	Defines indirect data (UC) Terminator of a binary operand (LC)
RSET	RSET	67	Reset the console to wait input status (error correction)
ALTR	α	70	Editing operator
"	"	71	Comments mode definition
SKIP	$	72	Unconditional branch
REP	:	73	Do loop generator (repeat)
✓	✓	74	Label sign
?	?	75	Conditional branch
→	→	76	Store data from AC
LEV	k	77	Operator or data level change request

Note:
UC = Operator (upper case)
LC = Operand (lower case)
KEY = Character representation of KEY
AC = Accumulator

APPENDIX B

1. Flow chart of INTERSPIKE INTERVAL OR AMPLITUDE HISTOGRAM PROGRAM:



Appendix B

2. LISTING OF PROGRAMS USED FOR INTERSPIKE INTERVAL AND AMPLITUDE HISTOGRAMS

INHS (O)

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

INH2 (P)

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

AMHS (4)

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

LODT (Z)

```
[illegible]
[illegible]
```

MNSG (*)

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

DSPL (8)

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

CUMM (C)

```
[illegible]
[illegible]
[illegible]
[illegible]

    [illegible]
    [illegible]
```

APPENDIX B

4. Symbolic Data References

a. Scalars

G = Mean of numbers in data block

U = Sum of mean squared of each block, $\sum_i G_i^2$

V = Cumulative mean squared

M = Sum of mean of each block, $\sum_i G_i$; at conclusion of processing contains cumulative mean.

P = Sum of variance of each block, $\sum_i \sigma^2$; at conclusion contains cumulative standard deviation.

b. Vectors

H = Cumulative histogram vector

B = Data vector

R = Working buffer

REFERENCES

Adey, W. R. Spectral Analysis and Pattern Recognition Methods for Electroencephalographic Data. Data Processing Conference, Copenhagen, April 1966, pp. 409-438.

Betyar, L. A User-Oriented Time-Shared On-Line System. Communications of the ACM, 1967, 10:413-419.

Feller, W. An Introduction to Probability Theory and Its Application. Wiley and Sons, New York, 1957, 2nd ed.

Levitan, H., Segundo, J. P., Moore, G. P. and Perkel, D. H. Statistical Analysis of Membrane Potential Fluctuations. Relation with the Pre-Synaptic Spike Trains. In Prep.

Moore, G. P., Perkel, D. H., and Segundo, J. P. Statistical Analysis and Functional Interpretation of Neuronal Spike Data. Ann. Rev. Physiol., 1966, 28:493-522.

Perkel, D. H., Gerstein, G. L. and Moore, G. P. Neuronal Spike Trains and Stochastic Processes. I. The Single Spike Train. Biophys. J., 1967, 7:391-418.

Rovner, L. D., Brown, D. and Kado, R. T. A Time-Shared Computing System for On-Line Processing of Physiological Data. Proc. Symp. on Biomed. Engin., Milwaukee, Wis. 1966, 1:153-156.

Rovner, L. D., Johnston, R. M., and Betyar, L. Event Detection During On-Line Processing. In Prep.

Scheer, A. L. An Analysis of Time-Shared Computer Systems. Research Monograph No. 36, The MIT Press, Cambridge, Mass., 1967, p. 115.

Walter, D. O. Rapid Interaction with a Digital Computer-Plusses and Minuses. The Physiologist, 1966, 9:313 (Abstract).

Figure 1. Diagram of Experimental Procedure and Data Form.

In the center is a representation of an isolated visceral ganglion of *Aplysia californica* with its connectives. One afferent fiber is shown entering the ganglion at the upper left, and reaching a cell which in turn sends an efferent process out of the ganglion at the lower right. Insert *a* illustrates a train of pulses delivered to the nerve by a stimulator. *b* shows the membrane potential fluctuations monitored by the microelectrode within the cell. After the third stimulus the membrane potential reaches the threshold potential and a spike is produced in the cell. *c* depicts the spike as it would appear in the axon of the cell in another connective.

Figure 2. Analog-to-Digital Conversion.

a) At the top an analog record of a spike train is shown, where the interval between consecutive spikes varies from 800 milliseconds to 100 milliseconds. This train was digitized using the event detection mode at a rate of 200 samples/second and stored on digital tape with the label BOk1. Below is a plot of the digital record for the analog train. It was recovered from the digital tape and displayed on the scope by the sequence of button pushes: LOAD BOk1° PLOT. The ordinate of the display is the length of consecutive intervals in numbers which are proportional to the sampling rate, and the abscissa gives the order of the interval. At right is a calibration for the length of the intervals.

b) At top is a sinusoidal function of 2.5 cps. When digitized at a rate of 200 samples/second and recovered from the tape by the instructions LOAD BOk2° PLOT it appears on the scope as the sequence of samples shown below. The ordinate of the digitized display is in numbers whose sign and size are proportional to the sign and size of the analog data, and the abscissa gives the order of the sample.

Figure 3. Interspike Interval Histogram and Autocorrelogram of Impulse Train.

a) Interspike interval mean (in seconds), standard deviation (in sec) and histogram for an impulse train of 1440 intervals generated by a Geiger counter exposed to a weak radioactive source. Number in upper right corner (.199) is the mean interval, in seconds, of the last 144 intervals in the train. Histogram is normalized so that the area under it is 1 and thus ordinate gives the relative frequency of the various interval lengths in the train. There are 144 bins over the range of 1.5 seconds.

b) Autocorrelogram for the same impulse train. There are 144 bins over the 15 second range of the autocorrelogram.

Figure 4. Amplitude Histograms of Common Periodic Functions.

Examples of histograms derived from periodic sine (a) and triangular (b) functions, generated electronically and digitized in the same way as the intracellular records. Each display shows a sample of the digitized potential as a function of time, running vertically, and above it the corresponding histogram. a) The sinusoid spends relatively more time at the extreme values than between peaks, so the histogram shows a mode at each extreme of the range. b) The triangular function is linear between the extremes and spends an equal amount of time at all potentials, thus yielding a uniform histogram.

Figure 5. Comparison of Effect of Regular and Irregular Excitation.

Membrane potential fluctuations of cell receiving a single excitatory input at a mean rate of 3/sec, regularly (top) and irregularly (bottom). The mean displacement of the membrane potential from its resting level was 10.7 mV under both types of stimulation. Scope display below compares the histograms associated with these oscillations. The relatively narrow and uniform histogram corresponds to regular excitations, the broad and skewed histogram is the result of Geiger-driven excitation. Histograms are normalized so that area under each is the same. Values of potential indicated by A and B in the intracellular records correspond approximately to the levels of potential so labeled in the histogram.

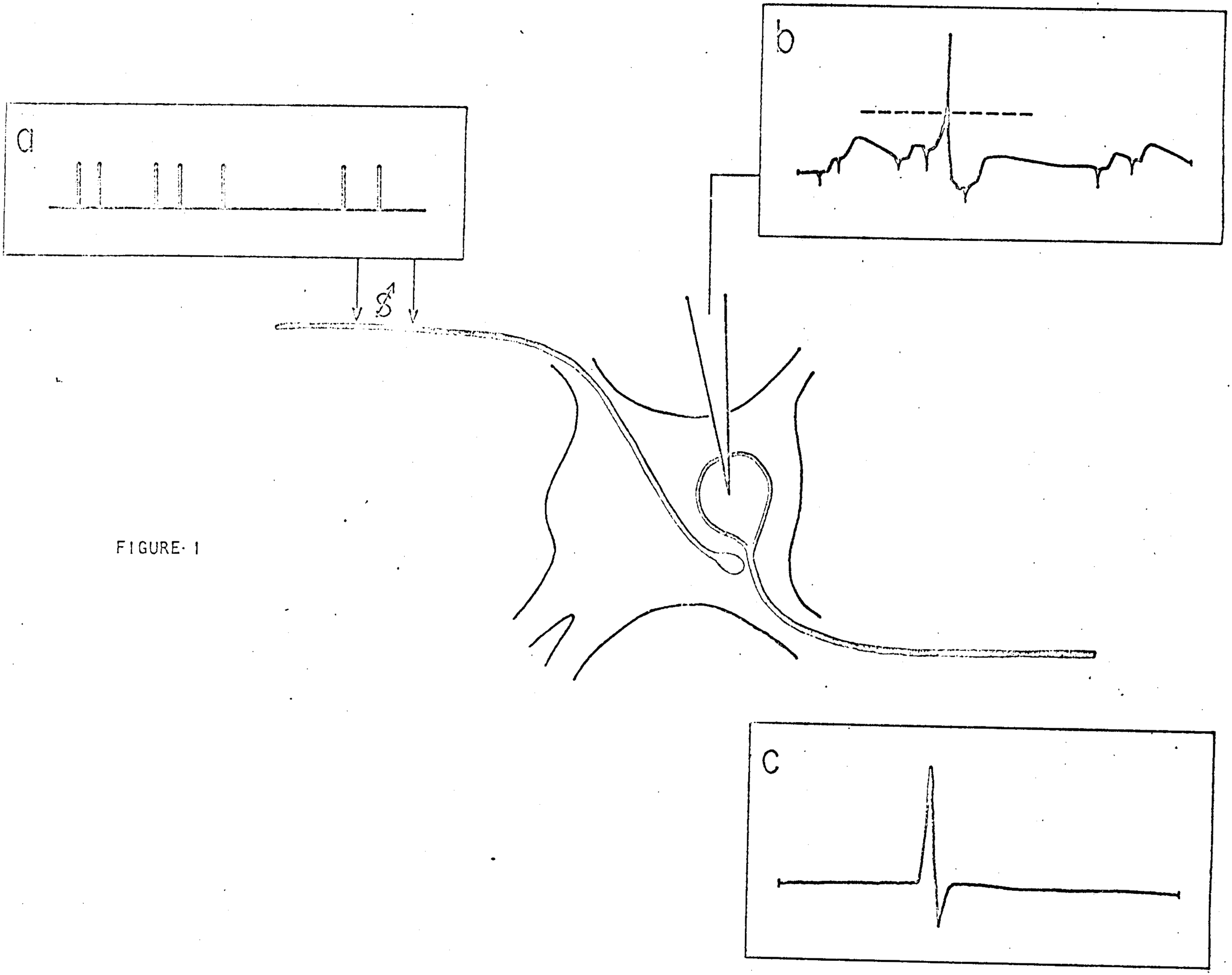


FIGURE 1

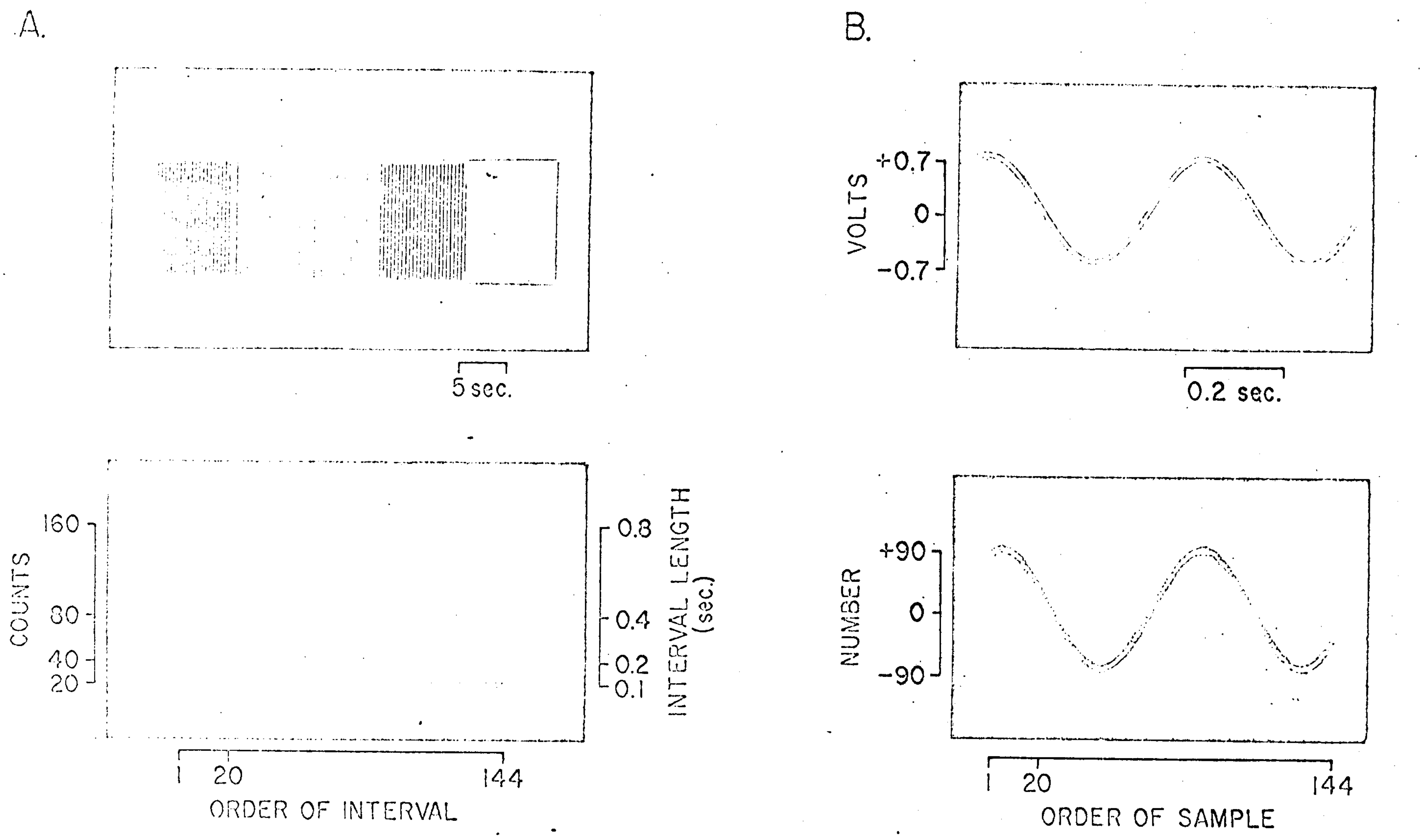

A.
5 sec.
COUNTS
160
80
40
20
INTERVAL LENGTH (sec.)
0.8
0.4
0.2
0.1
1
20
144
ORDER OF INTERVAL
B.
VOLTS
+0.7
0
-0.7
0.2 sec.
NUMBER
+90
0
-90
1
20
144
ORDER OF SAMPLE

FIGURE 2

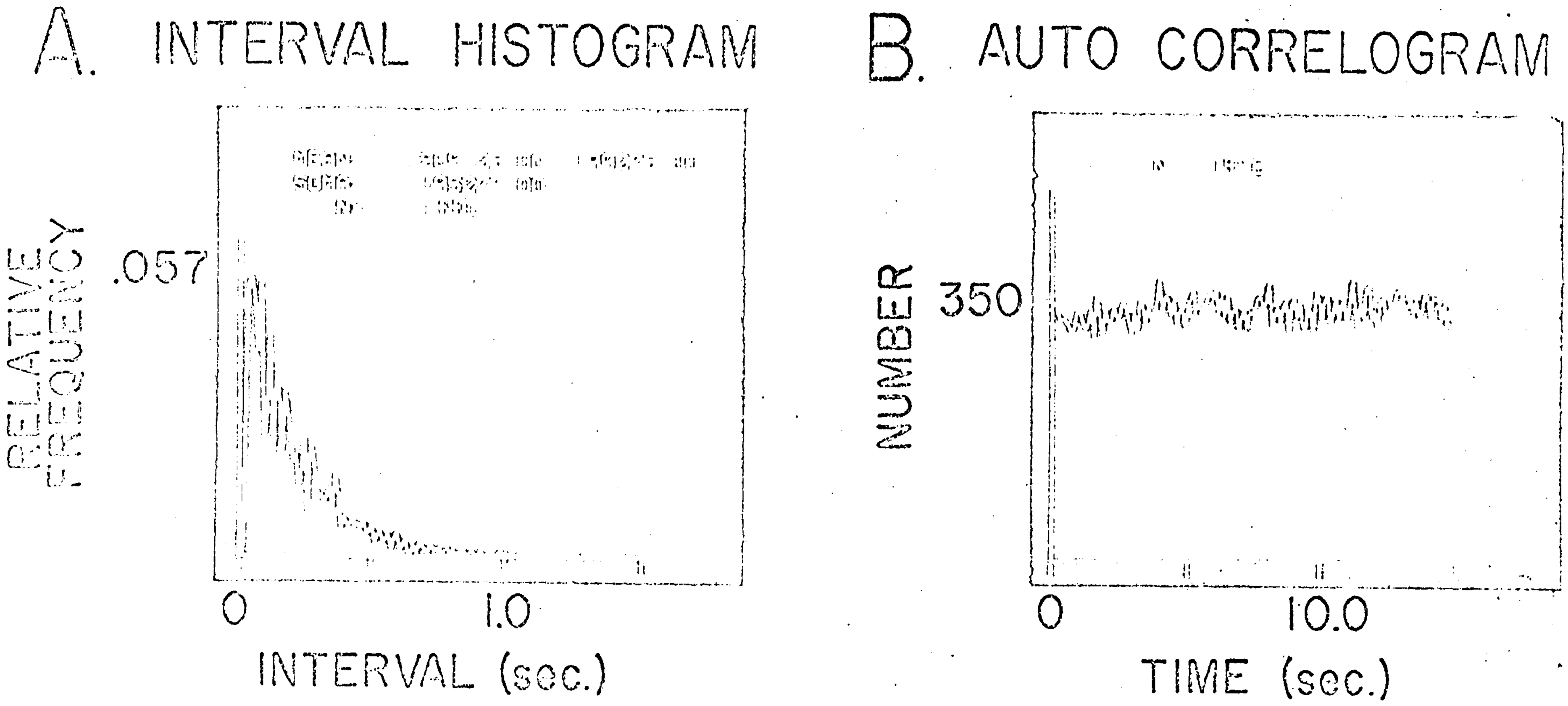
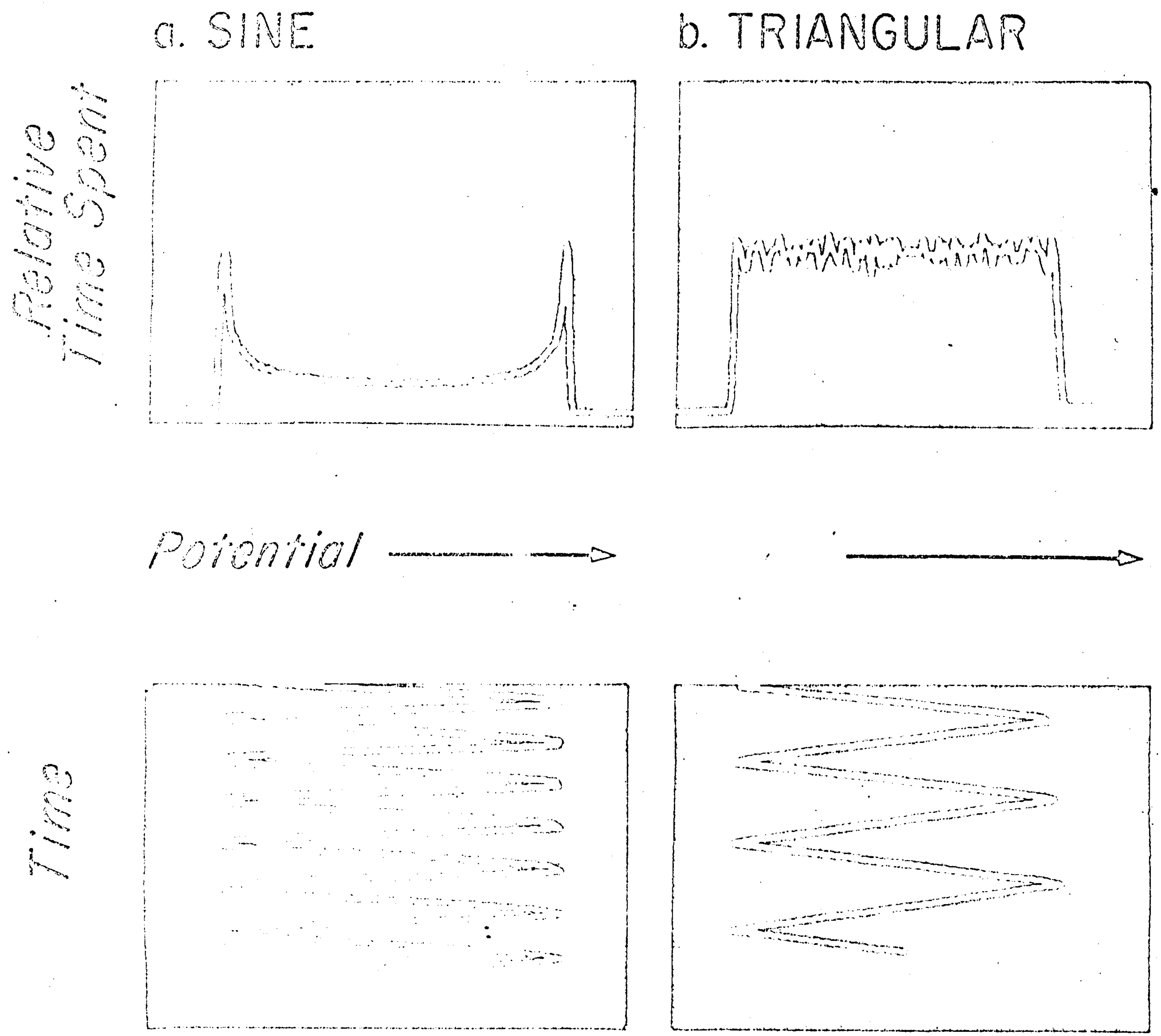

A. INTERVAL HISTOGRAM
RELATIVE FREQUENCY
.057
0
1.0
INTERVAL (sec.)
B. AUTO CORRELOGRAM
NUMBER
350
0
10.0
TIME (sec.)

FIGURE 3



FIGURE

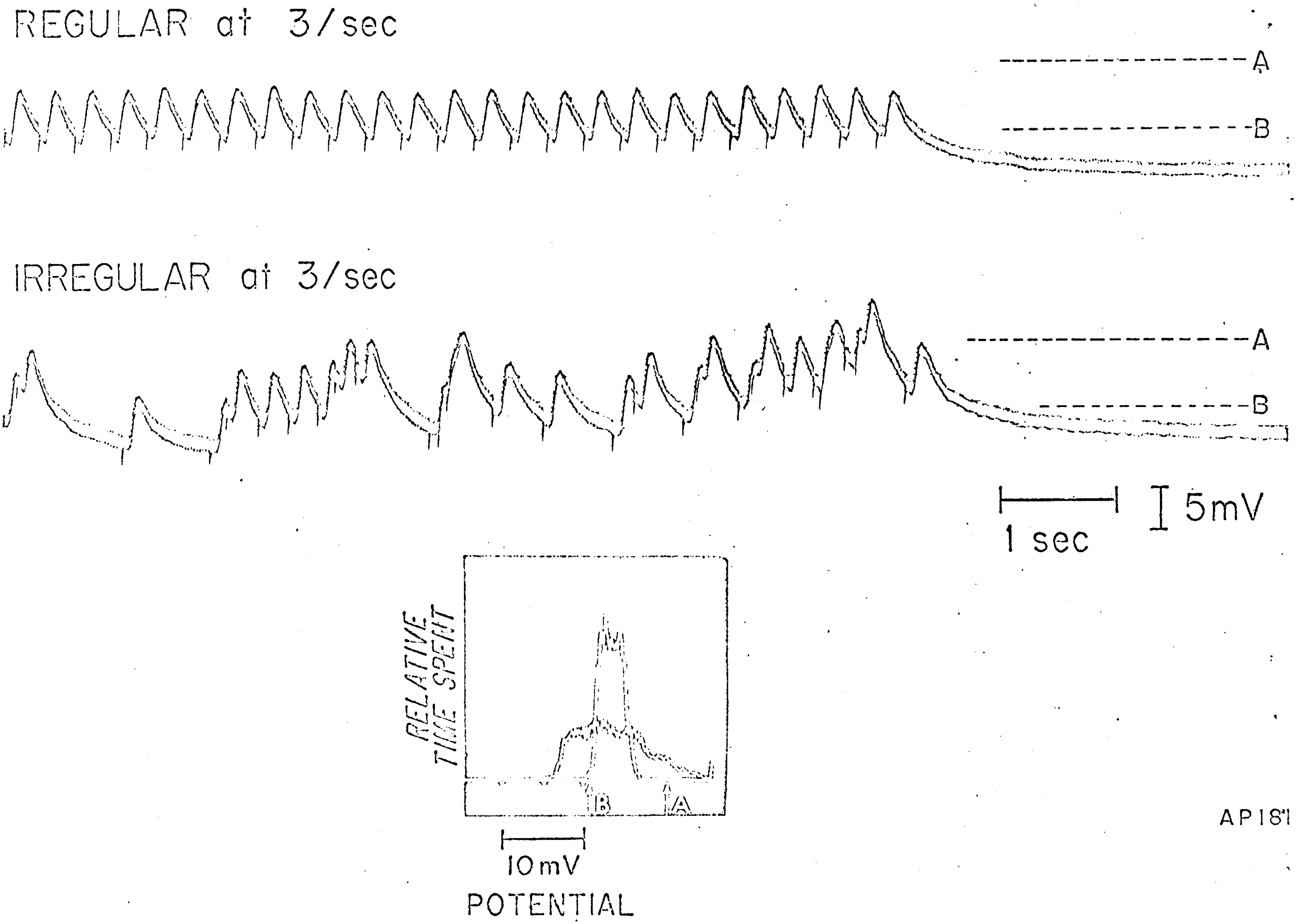


FIGURE 5

AP18'1