

Chapter 9

Computers

ARMAND B. CACHELIN, JOHN DEMPSTER and
PETER T. A. GRAY

*Hardware is the parts of a computer you can kick.
Computers don't save time, they redistribute it.*

This chapter is a short introduction to computers and their use in the laboratory. Not surprisingly, this chapter has become far more outdated than others in the six years since the publication of the first edition of *Microelectrode Techniques - The Plymouth Workshop Handbook*. The potential role of desktop computers was just beginning to emerge in 1987. Six years on, two computer generations later, the personal computer reigns undisputed in the office and in the lab. Thus the aim of this chapter is not so much to familiarise readers with PCs than to draw their attention to important points when choosing a computer or commercially available software for their lab. The chapter starts with a brief outline of some of the most important issues to consider when selecting a system. This is followed by sections which provide a more detailed consideration, firstly, of the basic hardware issues relating to the acquisition and storage of signals on a computer system and, secondly, of the functional requirements for software used for the analysis of electrophysiological data.

1. Selecting a system

Selection of a computer system when starting from scratch requires balancing a large number of factors. In general a guide-line widely accepted when specifying commercial systems, which applies as well here, is that the first thing to choose is the software. Choose software that will do what you want, then select hardware that will run it. In many cases, of course, the hardware choice may already be constrained by the existence, for example, of a PC of a particular type that must be used for cost reasons. However even in these cases the rule above should be applied to selection of other parts of the hardware, such as laboratory interfaces.

When selecting software it is essential to bear in mind likely future requirements as

A. B. CACHELIN, Pharmacology Department, University of Berne, 3010 Berne, Switzerland

J. DEMPSTER, Department of Physiology and Pharmacology, University of Strathclyde, Glasgow G1 1XW, UK

P. T. A. GRAY, 52 Sudbourne Road, London SW2 5AH, UK

well as current requirements. This can be very hard to do in any environment, as the introduction of a computer system always changes working patterns in ways that are hard to predict in advance. In a scientific laboratory, where the nature of future experiments is hard to predict anyway, this is doubly true. This is one reason for trying to select an adaptable system if at all possible. Maximum flexibility is obtained by writing your own software, but the cost in terms of time is high, and cannot generally be justified where off-the-shelf packages are available. However, if you are developing novel techniques it may be essential to have total control over your software. For most users applying established techniques, the choice is best limited to off-the-shelf systems, but the flexibility of the system should be a primary factor in the decision.

Flexibility in an off-the-shelf system can be assessed in many ways. For example, at the level of data acquisition are maximum sample rates, sample sizes, available complexity of output control pulses (e.g. for controlling voltage jump protocols) likely to match future requirements. At the next stage, factors such as the forms of data analysis available must be considered. How flexible are they? In many systems inflexibility here can often be partially overcome by exporting data to an external analysis package, provided that the software selected allows for data export in a suitable form. At the highest level of flexibility, can the system be used to control, sample and analyse data from completely new types of experiment? In most cases the answer, unsurprisingly, will be no. In this case, one must simply wait for off-the-shelf software to become available unless one writes the program oneself.

PART ONE: HARDWARE

2. Of bytes and men

Before going further it is worth introducing some basic concepts and terms which are widely used in computer-related discussions. Decimal numbers are a natural form of representation for ten-fingered humans. According to this system, numbers are composed of digits which take values between 0 and 9. Digital computers make extensive use of a different, so called binary number system. The elementary piece of information used in the computer is the bit which can take only two values (0 or 1). This binary system fits the ON-OFF nature of the computer's digital logic circuits. A number expressed in binary form consists of a string of '1's and '0's. For instance, the decimal number 23 ($2 \times 10^1 + 3 \times 10^0$) has the binary form 10111 which is equal to $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$.

Computers use a 'language' in which all words have the same length. Thus both operands (data) and operations (instructions) used by computer systems are encoded as fixed length binary numbers. Common word lengths are 16, 32 and more recently 64 bits. An 8 bit word is known as a byte and the term is often used as a unit of data storage capacity. One kilobyte (KB) is equivalent to 1024 bytes (2^{10} bytes)¹, a MB to 1048576 (2^{20}) bytes. The address bus width sets the limits to the address space. The

¹By convention, we shall use "K" as abbreviation for 1024 and "k" for 1000.

wordsize sets the width of data that a processor can manage conveniently. For example, 2^{16} (64 KB) is the largest number of memory locations that can be addressed directly by a 16 bit wide address bus. Additional bits must be used to store and fetch data from larger RAM. Of course 32 bit processors are much less limited being able to address up to 4 GB of RAM.

A further consequence of the use of a fixed length binary number system is that, at the most fundamental level, a digital computer can only store and manipulate integer numbers (0, 1, 2,...), so the highest integer number that could be stored within 16 bits is 2^{16} . The computer reserves one bit to represent the sign of a integer. Thus whole numbers in the range $\pm 2^{15}$ can be stored within 2 bytes. Although almost any kind of data can be stored and processed by a computer system, some forms of data are easier to handle than others. Large numbers ($>2^{15}$) and real numbers (i.e. with fractional parts, e.g. 1.23) cannot be directly stored in memory locations. How this and other problems are solved is the subject of the next section.

3. Computer systems

The digital computer forms the core of the data acquisition system, providing the following three basic functions: data input, processing and output. In the context of a laboratory these functions could be for example:

- Input: acquisition of signal(s) (analogue to digital conversion)
reading back data files
input of commands by keyboard or mouse
- Processing: qualitative signal analysis (preparation of 'idealised' data)
quantitative signal analysis (statistics, curve fitting, estimation of parameters of models)
- Output: command signals (digital to analogue conversion)
visual display of stored data
storage to data files on disk or tape
production of hard copies on paper
connection to other computer systems via a network

The advances in computer technology are such that most readily available personal computers within the IBM personal computer ('PC'²) and Macintosh families are more than adequate for the above tasks, in terms of computing speed and graphical display quality. Given the pace of change in computer technology, it is futile to provide particular specifications as they must become rapidly obsolete. In the following, we present certain general observations which should be borne in mind when choosing a computer.

Central processor unit

As can be seen from Fig.1, the central processing unit (CPU) forms the core of the

²From now on and throughout this chapter we will use the term 'PC' to designate a computer that is compatible with the IBM PC family of computers regardless of the model (PC/XT/286, 386...) or manufacturer (IBM, Compaq, Hewlett-Packard etc.).

computer system. It is, essentially, a machine for processing binary numerical data, according to the instructions contained in a binary program. The operation of the CPU is governed by a master clock which times the execution of instructions, the higher the clock rate the faster the CPU³. The performance of a CPU also depends upon the size of the data that a single instruction can handle, calculations being more efficient if they can be performed by a single instruction rather than a group of instructions. Typical personal computer CPUs (c. 1993) can handle 32 bit numbers in a single instruction and operate at clock rates of 25-66 MHz. The two major personal computer families (PCs and Apple Macintosh) are based upon different CPU designs, PCs use the 80x86 family of Intel processors (or clones by AMD or Cyrix) while the Macintosh uses the Motorola 680x0 family. Over the years, each CPU family has evolved with the introduction of newer and faster CPUs, while retaining essential compatibility with earlier models. Currently used members of the Intel CPU family include the 80386, 80486 and the recently introduced Pentium ('80586'). Macintosh computers are built around the Motorola 68030 and 68040 processors. The first computers built around the PowerPC 601/603 chips (see BYTE article) will be available by the time this book is printed. The clear trend towards graphical user interfaces makes it advisable to choose a computer system which uses at least an 80486 processor. Modern CPUs have more than enough power for most PC applications. However, certain tasks, such as program development, lengthy numerical analysis, such as curve fitting, and graphics intensive applications, do benefit from the fastest possible processor.

Floating point unit

The arithmetic operations within the instruction set of most CPUs are designed to handle integer numbers of a fixed size. Scientific calculations, however, require a more flexible number system capable of representing a much wider range of numbers including the complete set of real numbers as well as handling trigonometric, transcendental and other mathematical functions. For this purpose, computers use an internal notation convention called 'floating point' in which real numbers are approximated as product of a power of 2 times a normalised fraction. Thus:

$$R = 2^k \times f,$$

where k is the exponent and f is the mantissa. The most common laboratory computers, PCs and Macintoshes, both represent these numbers internally following the IEEE standard 754. In both cases, ignoring details of ordering of bytes in memory, these are stored internally as follows:

←most significant bit	least significant bit→
S	exponent
significand	

The sign bit S is set if the number is negative. The exponent may be either positive or negative (see Table 1), however, it is internally stored as a number in the range

³This is of course only true when comparing identical CPUs. Different CPUs running identical clock rates need not perform the same operations at the same speed.

Table 1. Characteristics of floating point numbers (IEEE 754)

Type	Size <i>bits</i>	Exponent		Significand <i>bits</i>	Representational range*
		<i>bits</i>	<i>range</i>		
Single	32	8	-126 to 127	23	1.2×10^{-38} to 3.4×10^{38}
Double	64	11	-1022 to 1023	52	2.3×10^{-308} to 1.7×10^{308}
Extended	80	15	-16382 to 16383	64	1.7×10^{-4932} to 1.1×10^{4932}

*Positive or negative numbers within these ranges can be represented.

between 0 and some positive number (whose value depends on the floating point format used). The IEEE 754 standard establishes 3 types of floating point number, which differ in the amount of memory they require for storage and the range and precision of the numbers that they can represent. These characteristics are enumerated in Table 1.

This increased complexity makes floating point calculations much slower than integer calculations, particularly if they have to be constructed from the basic CPU integer arithmetic operations. However, the floating point performance of a computer can be greatly improved by using a floating point unit (FPU; also called a maths coprocessor). This is a device dedicated to the performance of floating point operations which works in close co-operation with the CPU, in effect adding a set of floating point arithmetic operations to the basic CPU instruction set. Floating point calculations performed using an FPU can be an order of magnitude faster than the same operations implemented in software using CPU instructions.

Computers in both the PC and Macintosh families can be fitted with an optional maths coprocessor. Intel 80x86 CPUs in PCs can be supplemented by an 80x87, a maths coprocessor, though computers based on the 80486DX and Pentium CPUs have these built into the CPU. Macintosh computers, based on the Motorola 68x00 family of CPUs can also be fitted with maths coprocessor; the 68881 is used with 68020 CPUs, and the 68882 with 68030 and some 68040 CPUs.

RAM memory

Random access memory (RAM) is the high speed integrated circuit memory used to store a computer program and its data while it is being executed by the CPU. The amount of RAM determines the size of program that can be executed and/or the amount of storage space available for data. At present, (c. 1993) a standard personal computer contains 4 MB of RAM. However, the requirements for RAM continue to rise, as increasingly sophisticated and, unfortunately also, larger programs are developed, so we recommend 8-16 MB RAM, which is likely to become the norm in the next few years.

Disc storage

Modern applications software, such as word processors, spreadsheets, etc., take up large amounts of disc space, some consuming as much as 15 MB of disc space. The storage of laboratory data also requires large amounts of magnetic disc space. A single two channel digitised record, containing 1024 samples per channel, requires 4 KB of disc space (2 channels, 2 bytes per sample). A day's experiment might produce

1000 of such records, occupying up 4 MB of disc space. The same amount of disc space will be taken up by less than 2 minutes of continuous sampling at 20 kHz on one channel. It is important, therefore, to ensure that there is sufficient disc storage space to comfortably accommodate the quantity of experimental work envisaged, particularly when the computer is being used as the only storage device for experimental signals. Overall, 200-300 MB of disc capacity should be available and for the more demanding laboratory applications, 1000 MB might be preferable.

The hard disk system should also allow rapid access. Software disk caches, which use some of the system's RAM are effective in speeding system response, but it is better, when buying new hardware, to ensure that the disk performance is as good as

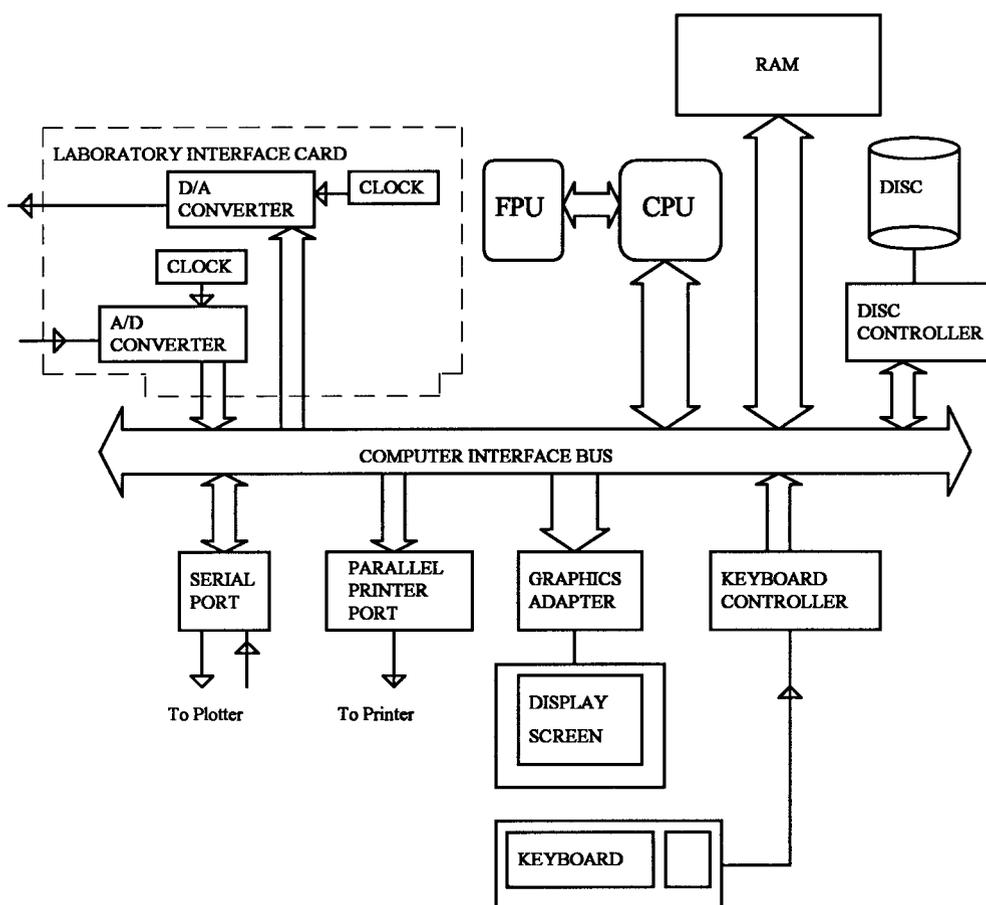


Fig. 1. Main computer components and sub-systems. Computer operation is governed by the central processing unit (CPU) which executes programs stored in the random access memory (RAM). Important sub-systems (graphics display, disc, keyboard) have their own semi-independent controllers. Data is transferred between the CPU, RAM, and these sub-systems, via the computer interface bus. The laboratory interface card, installed in an expansion slot, is connected to this bus and thus can exchange data between itself and RAM, and respond to instructions from the CPU.

can be afforded. A fast disk speeds up many applications more than a faster processor. On PCs a range of disk drives is available. The choice of disk interface may play a significant role in determining disk drive performance. The most common form is the IDE interface which is standard on many systems. SCSI drive interfaces generally offer better performance than other types, in particular for large drives. Performance may also be enhanced by using interfaces which have on-board memory, which is used to cache recently accessed data. Better performance still can be achieved on modern PCs with a local bus version of these cached interfaces, which pass data to the processor by a direct high speed link, instead of using the much slower general purpose interface bus that older models do.

Archive storage

Although it is convenient to store experimental data on the computer's internal hard disc, even the largest discs cannot accommodate more than a small number of experiments. Some form of removable data storage medium is therefore required for archival storage of the data. The secure storage of experimental data is an important issue. Experimental recordings are extremely valuable, if one takes into account the cost of person-hours required to design and perform the experiments. At least one 'back up' copy of the recorded data should be made on a different medium, in case the hard disc fails. It may also be necessary to preserve this archive data for many years, for scientific audit purposes or to allow future re-evaluation of results.

A variety of removable computer tape and disc media are available for archive storage. This is one area where rapid improvements in technology (and the consequent obsolescence of the old) does not help since it becomes difficult to determine which media are going to remain in widespread use. However, at present there are a number of choices. A number of tape forms exist, using cartridges of varieties of sizes and capacities (DC300, 45 MB; DC600, 65 MB; DAT (Digital Audio Tape) 1200 MB). Removable magnetic discs exist (e.g. the Bernoulli disc) but these are relatively expensive and have small capacities (40-80 MB) compared to alternatives.

In general, high capacity removable disc media use one of a number of optical data storage technologies. WORM (Write Once Read Many) discs, for instance, use a laser beam to vaporise tiny pits in the reflective surface of a metal-coated disc, which can be read by detecting the reflections of a lower-power laser light trained on the surface. As the name suggests, data can be written to a WORM disc only once, but read as many times as necessary. Rewritable optical discs are also available, which use the laser beam to effect a reversible phase change in the disc surface rather than a permanent pit. Two sizes of optical disc are available, 5.25 inch with a capacity of 640 MB (e.g. Panasonic LF-7010), and 3.5 inch supporting 128 MB (e.g. Sony S350 or Panasonic LF3000). The 3.5 inch discs are rewritable while the 5.25 inch drives support either WORM or rewritable discs.

A strong argument can also be made for storing a complete recording of the **original** experimental signals on some form of instrumentation tape recorder, designed for storing analogue signals. FM (Frequency Modulation) tape recorders, such as the Racal Store 4, with a DC-5 kHz recording bandwidth (at 7.5 inches per

second tape speed) have been used for many years. Analogue signals can also be stored on standard video cassette recorders, using a PCM (pulse code modulation) encoder to convert the signals into pseudo-video format. The Instrutech VR10B, for instance, can be used to store two DC-22.5 kHz analogue channels (or 1, and 4 digital marker channels) on a standard VHS video tape. These devices may gradually be replaced by modified DAT (Digital Audio Tape) recorders, such as the Biologic DTR1200 or DTR1600 which provide a similar capacity and performance to the PCM/video systems. DAT tape recorders are simple to operate compared to most computer systems and have a high effective storage capacity (a single 60 min DAT tape is equivalent to over 600 MB of digital storage)⁴. They provide additional security in case of loss of the digitised data, and allow the data to be stored in an economical and accessible form (compared to the non-standard data storage formats of most electrophysiology software). They thus provide the opportunity for the signals to be replayed and processed by different computer systems. An additional advantage is that a voice track can often be added during the experiment, offering a valuable form of notation. It **MUST** be borne in mind that **ALL** forms of magnetic tape (i.e. also DAT recorders) have a limited bandwidth though the reason for this limited bandwidth differs from one type of tape recorder to the other.

Networks

Laboratory computer systems are often connected to local area networks which allow data to be transferred between computers systems at high speed, permitting shared access to high capacity storage discs and printers. Often, the network can also be used to exchange electronic mail, both locally and, if connected to a campus network, internationally. Networks can be categorised both by the nature of the hardware connecting computers (e.g. Ethernet) and the software protocols (Novell Netware, Windows, NetBios) used to control the flow of data⁵. Within the PC family, the most commonly used network hardware is Ethernet followed by Token Ring. Ethernet permits data to be transferred at a maximum rate of 10 Mbps (bit per second). An Ethernet adapter card is required to connect a PC to an Ethernet network. Low end Macintosh computers are fitted with a LocalTalk network connector as a standard feature. However, LocalTalk, with a transfer rate of 230 kbps, is noticeably slower than Ethernet. On the other hand, high end Apple computers (Quadra family) and printers (Laser Writer IIg, Laser Writer Pro 630) are factory-equipped with serial, LocalTalk and Ethernet ports.

A wide variety of network software is available from a number of competing suppliers, the most commonly used being Novell Netware, PC-NFS (a system which allows PCs to use discs and printers on UNIX workstations), Windows for Workgroups on PCs or Asanté (and other programs) on Apple networks. Netware and PC-NFS are server-based systems where a dedicated file server is required to provide

⁴Two types of DAT recorders are currently available: "straight" data recorders and file-oriented back up devices. The data format and hence the access mode and time to the data is different.

⁵This is a little oversimplified since for example Ethernet is also a data transfer protocol (IEEE 802.3).

shared discs and printers. Windows for Workgroups, can act as a peer-to-peer network where any computer on the network can allow its discs and printers to be shared by any other. Macintosh computers also use a peer-to-peer system. In general, peer-to-peer systems are easier to set up, and are well-suited to the needs of a small group of 4-10 users within a single laboratory. Server-based systems, on the other hand, are better suited to the needs of large groups (10-100) on a departmental or institutional scale. They are complex to set up and administer, but have many features needed to handle electronic mail, archiving procedures, security, and remote printers which the simple peer-to-peer systems lack. Such networks require professional management, and large academic and research institutions may provide central support, particularly for Netware and PC-NFS. Differences between client/server and peer-to-peer are becoming blurred as time passes. Windows for Workgroups, for example, can access data on Novell file servers and can coexist with PC-NFS.

One final issue concerning networks is that there is often a potential conflict between the needs of the network adapter card and the laboratory interface unit. Particular care must be taken to ensure that the boards do not attempt to use the same DMA channel or interrupt request (IRQ) line. A more difficult problem, with running network software on a PC, is the 50-100 KB of RAM memory taken up by the network access software. The program has to be resident in the memory all the time leaving only 400-500 KB free memory (DOS limits the amount of memory for running programs to the lower 640 KB) which may not be sufficient to load some of the more memory-demanding data acquisition programs currently in use. DOS 5.xx (and above) by allowing loading of device drivers and short programs in the so-called 'high' memory (i.e. the address range between 640 KB and 1 MB) frees room in the 'low' memory for running other programs. These memory limitations problems are not known to Macintosh computer users because directly addressable memory is not limited by the operating system. Furthermore (slow) local network support for up to 32 users is provided from System 7.xx on. On PCs memory limitations are likely to become a thing of the past for good with the introduction of 'Windows 4' (currently code-named Chicago, which could be the end of DOS) or are already abolished by Windows NT.

However, while it may be advantageous to have a laboratory computer linked to a network (for example, to allow for exchange of data, text and figures between PCs), in general, timing constraints mean that it is not possible for a PC to act as a server during data acquisition or experimental control.

4. Data acquisition

Acquisition is the most important operation performed on the experimental data. It transforms the original analogue experimental signal into a data form (binary digits) that can be handled by a computer. Some of the relevant steps are discussed below. A general introduction to data acquisition and analysis can be found in Bendat and Piersol (1986).

Analogue to digital conversion

The outputs from most laboratory recording apparatus are analogue electrical signals,

i.e. continuous time-varying voltages. Before this analogue signal can be processed by a computer system it must be brought into a computer-readable form. A digitised approximation must be made, by using an analogue to digital converter (ADC) which takes samples (i.e. measurements) of the continuous voltage signal at fixed time intervals and quantifies them, i.e. produces a binary number proportional to the voltage level, as shown in Fig. 2. The frequency resolution of the digitised recording depends upon the rate at which samples are taken. The signal to noise ratio and amplitude resolution depend on the number of quantization levels available (machine-dependent) and actually used (signal preparation!). The choice of the optimal sampling frequency and the importance of the resolution are discussed in Chapter 6 (see also Colquhoun and Sigworth, 1983; Bendat and Piersol, 1986).

The choice of commercially available ADC performances is very broad with resolutions available between 8-20 bits and sampling rates as high as 1 MHz (1 Hz = 1 sec⁻¹). However, for general use within the laboratory, an ADC with a sampling rate of 100 kHz and 12 bit resolution, allowing the measurement of 2¹²=4096 discrete voltage levels, with ±5 volts input range⁶ is quite sufficient (though ±10 V might better match the output range of certain voltage-clamp amplifiers, see Table A1).

ADC facilities can be added to most personal computers by installing a specialised electronic device. This is either an internal circuit card or a specialised external laboratory interface unit. Either form supplies the ADC, digital-analogue converter (DAC) and timing hardware necessary to allow the computer to interact with laboratory data and equipment. The circuit card is usually installed in one of the slots of the computer expansion bus (see Fig. 1). A data 'bus' is actually more a data highway with multiple lanes used to transfer data between CPU and computer peripherals. An external laboratory interface unit is connected to the computer via a specialised interface card which sits in the expansion bus. Software running on the personal computer can thus both control the operation of the laboratory interface and data transfers to and from computer memory.

Analogue signal conditioning

Matching the data to the ADC

It is important to ensure that the voltage levels of the analogue signal are suitably matched to the input voltage range of the ADC. A number of ADCs are designed to handle voltage ranges of ±5 V. With 12 bits of resolution, the minimum voltage difference that can be measured is 2.44 mV (10000 mV/4096). However, many laboratory recording devices supply signals of quite small amplitude. For instance, a microelectrode amplifier (e.g. World Precision Instruments WPI 705) recording miniature synaptic potentials from skeletal muscle might provide signals no more than 1-2 mV in amplitude at its output. If such signals are to be measured accurately, additional amplification (e.g. ×1000) is required to scale up the signals so that they

⁶Signal input can be either single-ended (se) or differential (di). The signal of a single-ended input is measured relative to ground. Alternatively using differential inputs the difference between two inputs is measured.

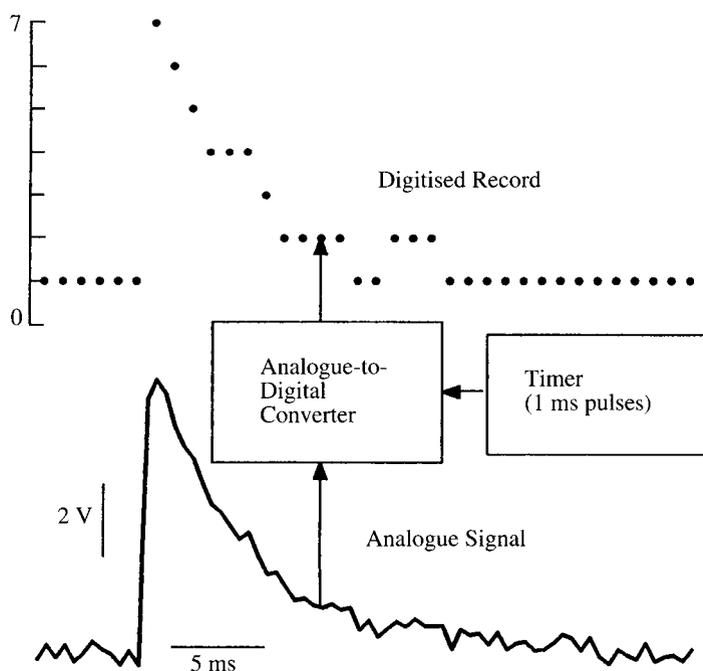


Fig. 2. The digitised record (upper trace) has been prepared from the continuous analogue signal (lower trace) by taking samples of the signal at 1ms intervals using a 3 bit A/D converter. The resulting record consists of 36 integer numbers with values in the range 0-7. Under normal circumstances, a much higher sampling rate and a 12 bit A/D converter would be used, allowing the values of individual data points in the digitised record to range from 0-4095.

span a significant fraction of the ADC voltage range. If the signals are superimposed upon a relatively large DC level, such as the -90 mV resting potential in the case of the mepps, it is also necessary to subtract this level off before amplification, in order to keep the signal level within the working range of the amplifier. A DC-offset facility is therefore required.

Filters

It may sometimes be necessary to filter the analogue signals to remove some of the high or low frequency signal components prior to digitisation and analysis. For example when performing noise analysis, all signal components with frequencies higher than half of the sampling rate **MUST** be removed from a signal in order to avoid aliasing of the power spectrum⁷. Low-pass filtering may, additionally, be required to improve the signal-noise ratio or to diminish the false-event detection rate (see Chapter 6 on single channel analysis). The low-pass filter used for this purpose should

⁷At least two points per cycle must be sampled to define a frequency component in the original data, so the highest frequency that can be resolved is half the sampling frequency (the so-called Nyquist or folding frequency). Frequencies above the Nyquist frequency in the sampled data will be folded back (aliased) into frequencies between 0 and the Nyquist frequency. In practice 5-10 points per cycle are better.

be of a design which provides a frequency response with as sharp a roll-off as possible above the cut-off frequency⁸, but without producing peaking or ringing in the signal. High-pass filters are less frequently needed, but are used when performing noise analysis to eliminate frequencies below those of interest. The roll-off characteristics of filters vary e.g. *Bessel*, used for single channel data and *Butterworth*, used for noise analysis. *Tchebychev* filters are not used for analysis but are found in FM tape recorders (e.g. RACAL) to remove carrier frequencies (see 'Filters' in Chapter 16).

Event detection

For some types of experiments, particularly those including random events, there may also be need for an event discriminator. This is a device which detects the occurrence of large amplitude waveforms within the incoming analogue signal. Typical uses include the detection of the signals associated with the spontaneous release of quantal transmitter packets (miniature synaptic currents or miniature synaptic potentials) at central and peripheral synapses or the opening of ion channels during agonist application. The detection of events by the software is discussed below. A hardware discriminator monitors the incoming analogue signal level and generates a digital TTL (transistor-transistor logic) pulse each time the signal crosses an adjustable threshold. TTL signals represent binary ON / OFF states by two voltage levels, 0 V (LOW) and about 5 V (HIGH). The TTL output of a discriminator is often held at 5 V, with a short pulse step to 0 V signifying that an event has been detected.

The TTL signal can be used to initiate an A/D conversion sweep by the laboratory interface to capture the event. In practice, it is not as simple as this since there is often a requirement for obtaining 'pre-trigger' data samples so that the leading edge of the waveform is not lost. Discriminators are also useful for matching trigger pulses from a variety of devices (e.g. Grass stimulators, Racal FM tape recorders) which produce signals incompatible with the TTL trigger input requirements of most laboratory interfaces. Suitable event discriminators include the World Precision Instruments Model 121, Axon Instruments AI2020A, Neurolog NL201 module or CED 1401-18 add-on card for the 1401.

All-in-one signal conditioner

In summary, a flexible signal conditioning system should provide one or more amplifiers with gains variable over the range 1-1000, facilities to offset DC levels over a ± 10 V range, a low/high pass filter with cut-off frequencies variable over the range 1 Hz-20 kHz and an event discriminator. Such systems can be constructed in-house with operational amplifier circuits (Discussed in Chapter 16). However, it is often more convenient to purchase one of the modular signal conditioning systems, available from a number of suppliers. Digitimer's (Welwyn, UK) Neurolog system, for instance, consists of a 19 inch rack mountable mains-powered housing, into which an number of differential amplifiers, filters and other modules can be inserted. Similar systems can be obtained

⁸The frequency at which the ratio of the output to the input voltage signal amplitudes is 0.7 (-3dB). 1 dB (decibel) = $20 \times \log(\text{output amplitude} / \text{input amplitude})$, see Chapter 16.

from Fylde Electronic Laboratories (Preston, UK). The Frequency Devices LPF902 low-pass Bessel filter unit can also be used as a simple signal conditioner, providing a differential amplifier with gains of $\times 1$, $\times 3.16$, $\times 10$, and an 8-pole Bessel filter with a cut-off variable over the range 1 Hz-100 kHz (noise analysis requires a Butterworth filter). The characteristics of certain commercial filters can be switched between Bessel and Butterworth and from low to high pass (e.g. EF5-01/02, Fern Developments).

Whatever signal conditioning system is used it is important that the exact amplifier gain and other factors are available to the computer software in order that the digitised signals can be related to the actual units of the signal being recorded (e.g. pA or μA for membrane currents and mV for membrane potentials). In the past, this has had to be done by the user manually entering calibration factors into the program after reading switch settings from the signal conditioners - a rather inconvenient and potentially error-prone exercise. Recently however, a number of suppliers have introduced computer-controlled signal conditioners, such as the Axon Instruments CyberAmp series and the Cambridge Electronic Design 1902, which can be directly programmed from the computer (usually via an RS232 serial port). In these devices, the amplifier gain, DC offset levels, and filter cut-off frequencies can all be directly programmed and read from the software running on the computer system. Such systems have the potential to greatly simplify the calibration of experimental recordings by allowing the computer to automatically monitor any changes to the amplifier gain and DC offset levels. Similarly, some modern voltage and patch clamps now have facilities to 'telegraph' their amplifier gain settings to computer systems, either using RS232 communications lines (Axon Axopatch 200, GeneClamp 500) or as fixed voltage levels (Axoclamp 2A). A particularly advanced example of this approach can be seen in the EPC-9 patch clamp (Heka Elektronik) where the controlling computer and patch clamp are closely integrated, allowing patch clamp parameter settings (including capacity and series resistance compensation), to be automatically controlled by the software.

Data and control outputs

In addition to sampling data it is also often desirable to generate output signals that may be used, for example, to control the experiment, to present a display of the sampled data on an XY monitor scope or to trigger the sweep of a conventional YT oscilloscope so that the display is synchronised with stimuli sent to the experimental preparation. Such outputs fall into two main classes, digital outputs and analogue outputs. Digital outputs are single bit digital signals, that can be either Off or On. These generally take the form of TTL signals. Signals like this can be used to provide trigger signals for oscilloscopes or to control two state devices such as taps. Analogue outputs are converted from an internal digital representation into an analogue signal by a digital to analogue converter (DAC). A DAC essentially performs the reverse transformation to that provided by an ADC, thus a number (between 0 and 4095 with a 12 bit DAC) held in the computer is converted to an output voltage that represents an appropriate fraction of the total output range, usually ± 5 or ± 10 V. Analogue

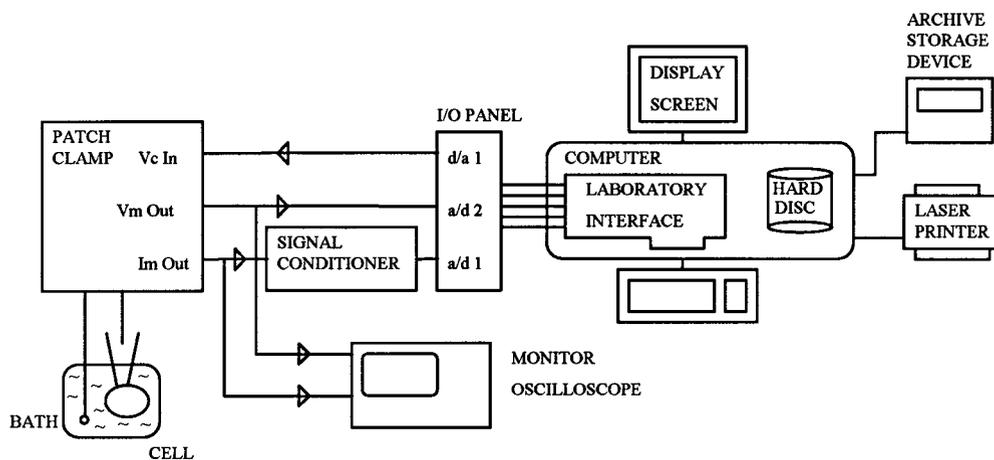


Fig. 3. Computer-based data acquisition and control system for an electrophysiological experiment. Cell membrane current (I_m) and potential (V_m) signals from a patch clamp amplifier are fed into A/D converter channels 1 and 2 of the laboratory interface, digitised and stored on the computer's hard disc. The current signal is amplified and low-pass filtered using a signal conditioning unit. Voltage command pulse waveforms can be generated by the computer and output to the patch clamp (V_c In) using a D/A converter. Hard copies of waveforms are produced on a laser printer and digitised signal records are archived on a removable disc or tape storage device.

signals can be used, for example, to control the holding potential of a voltage clamp amplifier, or to plot a hard copy of the incoming data to a chart recorder.

The laboratory interface

A laboratory interface used for electrophysiological data acquisition should have at least the following specification:

- 8 analogue input channels
- 12 bit resolution
- 100 kHz sampling rate (with hardware provision to enable sampling to disk)
- direct memory access (DMA) data transfer
- 2 analogue output channels (3 if a scope is to be driven by DACs)
- independent timing of analogue to digital conversion and digital to analogue waveform generation
- 4 digital inputs and 4 digital outputs⁹
- interrupt-driven operation¹⁰

A wide range of laboratory interfaces, meeting most of these specifications, are

⁹Though often one finds 16 digital input / outputs due to word length.

¹⁰Certain computer actions (sampling) should not be held up by any other computer operation. Precisely timed actions are obtained using so-called interrupt routines. Devices (eg a clock) issue an interrupt request - a little like the telephone ringing. The CPU acknowledges the interrupt request, saves the current program location and starts the interrupt routine (just like answering the phone) at a predetermined priority level. Upon completion of the interrupt routine, program location is restored and normal activity resumed.

available for both PC and Apple Macintosh families of computers. In the electrophysiology laboratory, the devices most commonly used in combination with PCs are the Scientific Solutions Labmaster DMA card, the Cambridge Electronics Design 1401*plus* interface (also for Mac) and - to a lesser extent - the Data Translation DT2801A card (Table A1 contains information on a more recent DT card). On the Apple Macintosh, the Instrutech ITC-16 is quite widely used.

The Labmaster card was one of the first laboratory interfaces to become available for PCs and has been in use for more than 10 years. Its popularity stems from its use with the pCLAMP software supplied by Axon Instruments. However, its old design limits its future even though it was updated in 1988 to support DMA - direct memory access, a hardware technique for transferring digitised data into the computer memory, essential for high speed data acquisition operations (Table A1 contains documentation on the follow up card, the Labmaster AD). Axon Instruments has now replaced it with an interface of their own design, the Digidata 1200.

The CED 1401 interface has also been in use for a number of years, particularly in Europe. It is a quite different design from the Labmaster card, being more a special purpose data acquisition computer in its own right, with an on-board CPU and RAM memory capable of executing programs and storing data independently of the host computer. The 1401 has recently been superseded by the much higher performance CED 1401*plus* which uses a 32 bit CPU rather than the original 8 bit 6502. An unusual feature of the CED 1401 is that it can be connected to a wide range of different types of computer system, including not only PC and Macintosh families, but also minicomputers such as the VAX and Sun workstations, via an IEEE 488 standard connection (a specialised parallel port).

Quite often the choice of laboratory interface is determined more by the choice of electrophysiological analysis software, rather than basic specifications. However, these specifications can be critical when writing one's own software. Unlike many other computer system components (e.g. disc drives, video displays), there are currently no hardware standards for the design and operation of laboratory interfaces. Each device has its own unique set of I/O ports and operating commands. Since most commercially available software packages work with only one particular interface card (usually manufactured by the supplier), it is not normally possible to substitute one type for another (an exception here is the Strathclyde Electrophysiology Software which supports several of the commonly available devices). The availability of suitable software has been one of the reasons for the widespread use of the CED 1401 and the Labmaster in electrophysiology labs.

5. Examining and exporting data

Once the data has been sampled, it must be examined and analysed. A video display unit is the minimum peripheral required for visual inspection. Permanent documentation requires some sort of hard copy device.

Video graphics display

Due to the lack of cheap high-resolution graphical display facilities in earlier computer systems (e.g. minicomputers in the 1980s), it was necessary to display digitised signals on an oscilloscope screen driven through DACs. Most modern personal computers are now capable of displaying data on the computer screen itself with enough resolution. Images are created on these screens by generating a bit map array in special video RAM in which each element determines whether a point on the screen is illuminated, in which colour (Red / Green / Blue) and at which intensity. The quality of the image is determined by the display resolution, measured as the number of independent pixels (picture elements) that can be illuminated. A typical screen display (c.1993) has a screen resolution of 640 (horizontal) \times 480 (vertical) pixels each pixel capable of being illuminated in 16 colours from a palette of 64. A resolution of 800 \times 600 is getting common and is more convenient for running Windows. Higher resolutions (up to 1024 \times 768 pixels, or more, and up to 16 million colours) are available but not usable by all software. In general a trade off must be made between higher resolution or more colours on one hand and both the speed and cost of the graphic display on the other.

With the increasing use of graphical user interfaces, the performance of the graphics display hardware has become at least as important as that of the main processor. In order to maintain adequate display rates, an increasing degree of intelligence has been built into video graphics cards, allowing basic operations such as the filling of areas of the screen with blocks of colour and drawing of lines, to be done by the display hardware rather than the CPU (see BYTE reference). The updating of a high resolution display also requires large amounts of data to be transferred from the main RAM into video RAM, making great demands on the interface over which data passes. In particular, the standard ISA (Industry Standard Architecture) interface bus found in most PCs, with its 8 MHz 16 bit data path, has become inadequate to support modern video displays. This problem has been overcome using special 'local bus' connectors, such as the VL or PCI buses, allowing data to be transferred to the video card over 32 bit paths and at much higher speeds. Many PCs now come equipped with both ISA interface slots and one or more local bus slots. The PCI bus is currently gaining acceptance for wider uses with its endorsement by IBM and Apple for use in their Power-PC based computers.

Static data display

Overall, the ability to combine text and graphics in colour, and its convenience, make the use of the bit mapped computer display preferable to using a separate oscilloscope screen, but important limitations should be noted. The 480 pixel vertical resolution of most screens is not sufficient to fully represent the 4096 levels of a 12 bit ADC sample. Similarly, the 640 pixel horizontal resolution limits the number of samples that can be displayed at one time, e.g. with a 20 kHz sampling rate, the screen is a mere 32 ms 'wide'. The displayed image is thus of much lower resolution than the digitised signal and does not contain (by far) as much information as is contained in the data itself. Brief/transient signals may be missed on a slow time scale. However, the lack of resolution can be partially overcome by using software with the ability to selectively magnify sections of the displayed signal.

Dynamic data display

The transient nature of the oscilloscope image, which persists only for a fraction of a second without refreshment, is well suited to the scrolling of signals across the screen, since only the new trace needs to be written. Images written to bit map, on the other hand, remain on the screen once written, supported by the graphics adapter display sub-system. This in turns means that to create a moving display the old trace must be erased (usually pixel by pixel because the fast screen erase erases ALL information from the screen) before a new trace can be drawn. Thus, implementing a fast scrolling display on a graphics display requires considerable handling of data and proves to be a difficult programming task. At present, few electrophysiological analysis programs have succeeded in implementing a scrolling display of the same dynamic quality as can be achieved using an oscilloscope. The problem is not insoluble, however, and it is likely that with the current rate of improvement in graphic display technology good scrolling signal displays will soon be realised on a computer screen. It is possible to generate scrolling displays using some existing hardware if the interface card is used to generate the X, Y and intensity ('Z') signals needed to generate such a display on an XY oscilloscope. For example, the support library supplied with the CED 1401 interface provides a command which allows programmers to generate such a display.

Hard copy graphics

The production of hard copies of stored signals, of sufficient quality for reproduction in a journal, is at least as important as the dynamic display of signals on screen. Hard copy devices fall into two main categories: printers which generate an image using a bit-map, like display screen graphics, and digital plotters which create the image by drawing lines (vectors) with a pen.

Until recently, the digital plotter was the most commonly used device for producing high quality plots of digitised signals. These devices, typical examples being the Hewlett Packard 7470A or 7475, are fitted with a number of coloured pens and can draw lines and simple alphanumeric characters on an A4 sheet of paper. Data is sent to the plotter in the form of a simple graph plotting language HPGL (Hewlett Packard Graphics Language) which contains instructions for selecting, raising and lowering pens, drawing lines etc. Although digital plotters can address over a thousand points per inch (1016 plotter units per inch) they position their pens to within a much larger fraction of a millimetre due to mechanical and other limitations (pens and penwidth). They can fully reproduce a 12 bit digitised record on an A4 page (10900×7650 addressable points). Plotters, however, are slow devices taking several minutes to produce a single trace with 1-2000 points. Also, the text produced by plotters was not quite of publication quality. Using HPGL has however the advantage that such files can be imported by a number of graphics programs (as will be discussed later).

Several laser printers can also emulate HPGL plotters¹¹, in particular the HP Laserjet III and IV series. Others can also do so provided they have been equipped with the

¹¹Availability of line thickness adjustment is an important prerequisite for publication-quality figures.

appropriate hardware or using emulation software running on the PC. This is a particularly useful feature since some (older) software packages only provide graphical output in HPGL form. To a large extent, plotters are being replaced by printers that can produce hard copy graphics of a similar quality to the plotter, and also very high quality text. The laser printer is ideally suited for general laboratory work, being fast, capable of producing high quality text in a wide variety of typefaces. Laser printers are also reliable and easy to maintain in operation.. The latest printers, with 600 dpi resolution, are capable of fully reproducing a 12 bit digitised record without loss of precision (7002×5100 dots for an A4 page). Ink jet printers (e.g. Hewlett Packard Deskjet, Canon Stylewriter/Bubblejet, Epson Stylus) provide a similar quality of text and graphics (300 DPI) but a lower cost. The most recent members of the HP Deskjet family make colour accessible at a low (Deskjet 550C incl. PostScript for just less than US\$ 1000.0) to moderate price (Deskjet 1200C incl. PostScript). They are still very slow (up to several minutes per page) and their use is recommended for final copies only.

As in many other areas of the computer industry, competitive pressures have led to the creation of de facto standards. The most important emulations are PCL5 (HP Laserjet III), PCL5e (HP Laserjet 4) and Postscript Level I or II (see BYTE November 1993 p. 276 ff). Other frequent emulations include ESC-P (EPSON LQ/FX), IBM Proprinter and HPGL. The various emulations are differentiated by the codes used to transmit formatting and graphics instructions to the printer. Postscript is a specialised Page Description Language (PDL) licenced from Adobe Inc. designed for the precise description of the shape and location of text characters and graphics on the printed page. One of its prime advantages is its portability. A Postscript description of a page of text should be printable on any printing device supporting PostScript PDL (not just a laser printer). It is in widespread use within the publishing and printing industry. The main practical difference between the Postscript and other graphic images is that Postscript's images are defined using vectors which are sent directly to the printing device while PCL graphics must be first created as bit-maps within host computer memory (a more complicated process) then sent to the printer.

In practice, there is little to choose between either type of laser printer. Postscript printers, perhaps, have the edge in terms of variety of type styles, while PCL printers are usually faster at printing bit mapped images. However, both of these families are widely supported by applications software such as word processors, spreadsheets and graph plotting programs. In addition, many printers now have both PCL and Postscript capabilities (e.g. HP Laserjet 4MP), while Postscript facilities can be easily added to older HP Laserjet printers using an expansion cartridge (e.g. PacificPage Postscript Emulator).

PART TWO: SOFTWARE

6. Operating systems

The operating system is a collection of interlinked programs which perform the basic operations necessary to make a computer system usable; reading data from the keyboard, creating, erasing, copying disc files, displaying data on the screen. The operating system

acts as a environment from which to invoke applications programs. It defines the characteristics of the computer system at least as much as the hardware does.

The operating system appears to the user as the set of commands and menus used to operate the computer. It acts as an interface between hardware and the user, hence the term 'user interface'. From the point of view of the programmer, the operating system is the set of subroutines and conventions used to call upon operating systems services - the applications programmer interface (API). The debate about the relative merits of different computer families (e.g. Macintosh vs. PC) is very much a discussion about the nature of the operating system and what features it should provide to the user, rather than about the hardware, which is essentially similar in performance.

Although this chapter is primarily concerned with the applications of the computer to data acquisition in the context of electrophysiology, a brief consideration of the development of operating systems for personal computers is worthwhile since they provide the basic environment within which our software has to function.

Character-based OS

Most electrophysiology software runs on PCs, and uses MS-DOS (Microsoft Disc Operating System). MS-DOS is a relatively basic, single task system; primarily designed for running only one program at a time. This simplicity of design, compared to more complex systems like the Macintosh OS or UNIX, has made it relatively easy to develop efficient data acquisition software for PCs. However, the simple nature of MS-DOS, which was well matched to the performance of computers of the 1980's, is increasingly becoming a major limitation in making effective use of the modern processors which are 100 times more powerful¹².

MS-DOS, like its ancestor CP/M, uses a command line interface (CLI), i.e. the user operates the system by typing series of somewhat terse commands (e.g. DIR C: to list the directory of files on disc drive C:). This approach, though simple to implement, requires the user to memorise large numbers of command names and use them in very syntactically precise ways. A number of utility programs (e.g. PCTools, Norton Commander, Xtree) simplify often used file operations (e.g. copy, move, delete) and launching of programs, as well as providing a simple, mouse-driven graphical user interface (GUI). A very primitive GUI called DOSSHELL is also present in DOS (since version 4). It enables starting several programs which can be run alternatively (but not simultaneously)¹³. Modern operating systems are expected to be able to run several programs at the same time and provide a GUI. Windows 4 ('Chicago') will eventually replace DOS (but emulate it).

¹²Most PC users have confronted, at least once, the 640KB limit on memory dictated by DOS and the architecture of the first PCs. Most PCs, with 386 and above processors, overcome this problem by using 32 bit memory addressing to support up to 4GB of memory. This feature (32 bit memory addressing) is supported by recent versions of Windows. Though this is also supported by so-called DOS extenders, we do not recommend this practice as it is likely to cause compatibility problems.

¹³It is of limited practical use. Windows (3.1 and above) is a much better alternative.

Graphical user interfaces

The major *external* difference between character-based and graphical user interfaces is the way the user interacts with the computer. Within a GUI, the user uses a hand-held pointing device (the mouse) linked to a cursor on the screen in order to point to pictorial objects (icons) on the screen, which represent programs or data files, or to choose from various menu options. The first widespread implementation of a GUI was on the Apple Macintosh family, and contributed markedly to the success of that machine. The concept has now been widely accepted and, with the introduction of the Microsoft Windows and IBM OS/2 for PCs, most personal computers are now supplied with a GUI.

Windows, OS/2 and Macintosh System 7 provide very similar facilities, being mouse-driven, with pull-down menus, 'simultaneous' execution of multiple programs linked to movable resizable windows. The main difference between these three systems is that on the Macintosh and in OS/2, the GUI is closely integrated with the rest of the operating system whereas the Windows GUI is separate from, but makes use of, MS-DOS. Apple, IBM and Microsoft are in close competition with each other to develop newer and more powerful operating systems for the 1990's. Microsoft, has recently released Windows NT, a new operating system which integrates the Windows GUI with a new underlying operating system, better able to support multi-tasking and without many of MS-DOS's limitations.

These newer operating systems also all provide some support for 32 bit applications, that is programs that can directly access the full 4GB of memory that the CPU can access. The degree of integration of this support varies, OS/2 and Windows NT providing full support, Windows 3.xx only partial support. The additional overheads involved in supporting the graphical user interface do, however, mean that there is a cost both in terms of performance and memory requirement in using these operating systems. The entry level requirements for hardware to run these operating systems (in terms of memory size and processor power) are greater than those for a basic DOS system: e.g. a DOS based system running on a 80286 processor and with 640 KB of memory would provide adequate performance for most tasks. In contrast the minimum possible system to take full advantage of the current version of Windows (3.1) is an 80386 processor and 2MB of memory. In fact even this configuration would quickly be found to be severely limiting if any use was made of the multitasking capabilities of Windows.

7. Electrophysiology software

Electrophysiology covers a wide range of experiments and produces many different types of data. The electrical signal produced by biological preparations in such experiments has mainly two components: time and amplitude. Thus, analysis of electrophysiological records is primarily concerned with monitoring the amplitude of a signal as a function of time, or the time spent at a given amplitude. Most electrophysiology programs are like a 'Swiss army knife': although they can deal with a lot of experimental situations, they usually fall short when the proper instrument is needed. The 'perfect program', which incorporates only the best subroutines to deal with each type of data obviously does not exist. This is usually

the reason why some expertise in programming is important. Writing one's own software may sometimes be essential to allow those experiments to be done that should be done and not those dictated by the software. It is the aim of the present section to mention which options should be present in a good electrophysiology program (see also Table A3-A5).

Most widely used electrophysiology packages for PCs (e.g. pCLAMP, CED EP Suite, Strathclyde Software) currently run under DOS. GUI based multitasking operating systems for PCs, such as Windows and OS/2 are gradually replacing DOS, though it may some time before character based OSs disappear altogether. This shift is reflected in a migration towards GUI versions of data acquisition programs: e.g. Axon Instruments' Axodata or Instrutech's PULSE for the Macintosh OS or CED's Spike2 for Macintosh, Windows 3.1 and Windows NT. Many standard applications (e.g. word processors, spreadsheets, graphic packages) are now available in a GUI-based version (e.g. for Windows, OS/2; Macintosh). How far electrophysiology packages follow the trend towards being GUI based depends to some extent upon the continued availability of character-based shells within future GUI OSs, such as the 'DOS box' within current versions of Windows and OS/2. There are undoubtedly great advantages that come along with GUIs, e.g. consistency of the user interface, ability to exchange data with other programs, multi-tasking¹⁴. Unfortunately, there is also a price to pay for the increased functionality of GUIs. It proves to be more difficult to develop programs for a graphical, multi-tasking environment than for a character-based OS. Fortunately, software publishers (e.g. CED) already provide direct-link libraries (DLLs) which greatly simplify communication with data acquisition hardware (e.g. 1401) or access to data files using proprietary file formats under Windows.

Different programs for different experiments

In the real world, most electrophysiology programs originate from teams which have designed their software to perform optimally with one type of experiment. Electrophysiology software has evolved in mainly two directions. Some programs were written to analyse short stretches of data often sampled in a repetitive way. Such programs were originally written to investigate the properties of voltage activated channels (e.g. sodium, potassium or calcium channels) in response to polarisations, such as voltage steps, applied by the experimenter. For example the AxoBASIC set of subroutines was uniquely tailored to perform repetitive tasks on a small number of data points.

Other programs have been designed to deal with random data at equilibrium; for example programs specifically conceived to sample and analyse long stretches of data obtained from agonist-activated ion channels. As already mentioned, the type of experiments that are likely to be done may often determine the choice of the program

¹⁴True multi-tasking, allowing several different programs to be active at once, by time sharing the CPU, is available in Windows enhanced mode. However, real time laboratory applications, such as sampling and experimental control, should normally be run exclusively, or there may be problems such as missed data points or timing errors.

and as a consequence the hardware e.g. requirements for interrupt driven data acquisition and large capacity data storage. It is important that experimental requirements determine software, and not vice versa; in early voltage-jump experiments hardware limitations in sample rates and durations had a profound effect on the information extracted from the data produced.

Fig. 3 shows the major components of a typical computer-based data acquisition system, as it would be applied to the recording of electrophysiological signals from a voltage or patch clamp experiment. The membrane current and potential signals from the voltage clamp are amplified and/or filtered via a signal conditioning sub-system and fed into the analogue inputs of the laboratory interface unit. Computer-synthesised voltage pulses and other waveforms can be output via DACs and applied to the command voltage input of the voltage clamp in order to stimulate the cell. On completion of the experiment, the digitised signals stored on disc can be displayed on the computer screen and analysed using signal analysis software. Hard copies of the signal waveforms and analysis results can be produced on a laser printer or digital plotter. As digitised signals accumulate, filling up the hard disc, they are removed from the system and stored on a archival storage device such as digital magnetic tape or optical disc.

Short events

The currents produced by voltage-operated channels during membrane depolarisation are typical short events lasting a few hundreds of milliseconds, though steady state may be reached only after much longer periods of time. It is important that the duration of each sample should be long enough for the relaxation of the properties of the channels to a new equilibrium to be complete during the sampling period. Essential program options are simple menus to adjust sample frequency and duration. Advanced program options include the ability to sample during several intervals within a repetitive pattern. The optimal sample frequency is determined by several parameters such as the required time resolution, and the filter-cut-off frequency is adjusted accordingly e.g. Chapter 6, Colquhoun and Sigworth (1983). Typically such experiments require sampling on two channels, e.g. voltage and current. Sometimes one may wish to monitor an additional signal such as the output of a photomultiplier tube, which allows measurement of the intracellular concentration of various mono- and divalent ions when used with appropriate indicators. It should be borne in mind that the maximum sample frequency (f_{max}) quoted for a laboratory interface applies to sampling on a single channel. The maximum frequency per channel when sampling on multiple channels is usually equal to f_{max} divided by the number of channels.

Watching the data on-line

One should **always** monitor the original, unmodified data on a scope. During repetitive sampling, this requires the software to keep triggering the scope even during episodes in which no data is being sampled (or manually switching the scope to a self-triggered mode). Also, it is often useful and sometimes essential to be able to watch the conditioned, sampled data on-line during acquisition.

Watching large amplitude signals such as whole-cell response to an agonist on-line

should not be problem provided the data has been correctly amplified prior to the acquisition (see Signal Conditioning above). The situation is entirely different if the relative amplitude of the data is small compared to the underlying signal. This is the case for single channel openings of voltage-activated channels. Rapid changes of the membrane potential are accompanied by capacity transients which can be much larger than the actual single channel amplitude. The amplitude of capacity transients can be at least partially reduced by adjusting appropriate compensatory circuits in the voltage or patch clamp amplifier.

The channel openings are also superimposed on a holding current whose size may be much larger than the amplitude of a single channel opening. Before details can be observed, it is necessary to subtract such offsets, usually due to (unspecific) leak currents and capacity transients, from the original data. If this cannot be performed by external hardware prior to or during acquisition, it is necessary to prepare a leak trace without openings to be subtracted from the sampled data. A good electrophysiology program should include options to do this on-line leak and capacity transient subtraction.

Leak current and capacity transient subtraction

A method often used for leak subtraction is the following. Let us assume that the channel is activated by stepping from a holding potential of -100 mV to a potential of -10 mV, a protocol often used to activate sodium channels. The individual channel openings will be much smaller than holding current and capacity transients. A correction signal is obtained by averaging N traces measured while inverting the command pulse thus hyperpolarising the membrane potential from -100 mV to -190 mV, a procedure which does not activate ion channels but allows measurement of the passive membrane properties, the silent assumption of this protocol being that these passive electrical membrane properties are purely ohmic, i.e. symmetrical and linear over a wide range of potentials. When the step potential is large, for example to -240 mV in order to compensate a step from -100 to $+40$ mV, a fraction of the voltage step is used instead (e.g. from -100 mV to -120 mV) and a linear extrapolation made.

Assuming that the leak current is linear, the sampled response is then scaled (in our example by a factor of 7) and may be used for correction of multiple voltage-step amplitudes ('scalable' leak in Table A3). This method was originally introduced by Armstrong and Bezanilla (1974) and called 'P/4' because the pulse amplitude was divided by 4. More generally the method is called P/N to indicate that an arbitrary fraction of the pulse is used. However, the assumption of linear leak is not always correct and the leak trace many need arbitrary adjustment ('adjustable' leak in Table A3). Most procedures for leak and capacity transient subtraction are not perfect so it is good practice to store both the original, unaltered data, as well as the leak subtraction and capacity compensation signal, so that a later, better or more detailed analysis can be performed as required. Another method of signal correction which, however, does not work on-line, consists in marking and averaging traces which do not contain single channel openings ('nulls') to prepare the leak and capacity

transient subtraction signal. This method has been used to analyse data obtained from voltage-dependent calcium channels.

It is important that the method used for preparation of leak and capacity transient compensation signals used by a software package should be clearly explained in the program documentation. Ideally, it should be possible for the user to modify the method used (for example by having access to the code).

Long samples

Some types of data require sampling for long periods of time. These are for example the slow responses of a cell following application of an agonist, or random data, such as the single channel openings or membrane current noise during agonist application or spontaneous miniature synaptic events.

Triggered events

Predictable events, such as evoked synaptic currents, are easy to deal with provided they have undergone adequate conditioning (e.g. amplification) prior to sampling. A number of programs exist which turn the computer into a smart chart recorder. These programs can record at moderate sampling frequencies on up to 16 and more channels. They include options for entering single character markers. These programs can usually play back the data but their analysis options are normally limited when they exist at all. Examples of such programs are CED's CHART, AXON's AXOTAPE or Labtech's Notebook programs. Their limited analysis capabilities make it essential that programs can export data in various formats for analysis by other software (see below).

Random events

Random events are more difficult to deal with. An important question is to decide whether all of the data should be recorded on-line as well as on an analogue recorder. The considerable amount of disk storage space required to store all of an experiment makes it worth considering detection hardware and algorithms in order to sample only the data which is actually relevant. Obviously, the event detector must be capable of keeping track of time between meaningful events (such as miniature synaptic events or single channel openings), as time intervals are one of the variables in the data. The event detector may be implemented in hardware, as described briefly above, or in software. Software event detectors sample data into an 'endless', circular, buffer. The size of the buffer determines how much data prior to the triggering event will be available.

A problem with the use of event detectors is that the experimental record between events is not seen. Signal properties, such as the noise levels, or the size of the event that fires the trigger, may change during an experiment. In the latter case, the event detector may fail to trigger and data will be lost. It is therefore essential if using this technique to record all original data to tape, so that re-analysis with, for example, different trigger parameters may be carried out if needed. In addition, the voice track on a tape recorder can contain far more useful information than a few comments typed quickly in a file.

Driving external devices

Several types of devices can be activated by a computer-generated signal. The simplest devices (taps, and other electromechanical equipment) require a simple TTL (on-off) signal which can be generated either via a DAC or parallel digital port. Triggering the monitor scope usually only requires a TTL pulse. One usually wishes to trigger the oscilloscope at a given time before say depolarising a cell so the response can be monitored during the entire duration of the depolarisation. A good electrophysiology program should allow triggering of external devices (oscilloscope, magnetic tap, amplifier, signal conditioner, etc.) at predetermined times.

The next level of complexity requires variable signals to be sent to a voltage clamp amplifier. Although the membrane potential or current level can be dialed manually on the voltage-clamp amplifier the command pulses need to coincide with data sampling so it is better to rely on the computer to generate voltage steps, though complex voltage-step protocols can also be generated by a programmable stimulator provided a trigger signal is provided to initiate sampling. In general, use of the computer to generate command pulses makes it easier to create a wide range of complex command signals such as voltage ramps, multistep protocols, and playback of pre-recorded samples, such as action potentials. Command signals are generated by a DAC. The amplifier is treated in these protocols as an entirely passive device, though the latest generation of patch clamp amplifiers, such as the Axon Axopatch 2000 and the List EPC9, are capable of two way communication with the computer.

At the highest level of sophistication smart devices such as digital oscilloscopes and digital plotters can interpret simple, human-readable, commands of alphanumeric data sent in ASCII form via a serial or parallel port. Examples are Hewlett Packard, and other HPGL compatible, plotters, some Nicolet and Tektronix oscilloscopes and the CED 1401, all of which use a proprietary command language, just as printers do.

Synchronous command pulses

Investigating the properties of voltage-activated ion channels often means using complex pulse protocols. For example, the inactivation of sodium, potassium and other voltage-activated conductances may be studied by use of prepulses applied before a test depolarisation to determine the proportion of channels that can be activated (see Chapter 2). The conductance of ligand gated channels, that show little time dependence can be tested with slowly increasing ramp commands to plot I/V curves directly. Alternatively, the conductance of voltage-gated channels may be investigated by applying a series of incrementing, or decrementing, pulses from a holding potential. We can see from these two examples that there should be menu options to prepare pulse protocols in advance.

For voltage-ramps, it is important that the program should maximize the number of voltage steps so as to minimize their size. The amplitude of the smallest V_{CMD} step is a function of both resolution and voltage range of the DACs: with 12 bit resolution and a voltage range of ± 5 V, one computer unit is converted to a V_{CMD} step of 2.44 mV which translates to a step of 49 μ V at the pipette tip after attenuation by the

amplifier (50-fold for Axon amplifiers). The size of the smallest V_{CMD} step is correspondingly larger (244 μV) with amplifiers which use a 10-fold attenuation (e.g. List). The latter size is such that each update of the amplitude (by one computer unit) will cause a small capacity transient and thus blur the current response. This problem can be minimised by low-pass filtering the DAC output before passing it to the voltage clamp amplifier V_{CMD} input.

Data processing

Processing electrophysiological data includes three operations: data preparation, statistical analysis and model fitting.

Data preparation

The aim of data preparation is to isolate those parts of the data that will be further analysed and to remove unwanted data stretches e.g. those containing artefacts or spurious signals. Editing a trace is accelerated if graphical tools (pointers, cursors), mouse support or a graphical user interface with pull-down menus, etc. are available. Although few of the currently available electrophysiology programs (e.g. Spike2 and PULSE for the Mac) provide mouse or true GUI support, it is likely that future releases of these programs will run under some form of GUI (Windows, OS/2 or Mac OS).

Some of the data preparation, such as baseline subtraction, capacity transient suppression and event capture may have already happened during acquisition. For slow responses the size and position of peaks or troughs of current as a function of membrane potential or other experimental parameters may be of interest. Alternatively, it may be desirable to isolate regions of a signal for analysis. Examples are fitting a sum of exponential curves to membrane currents obtained during prolonged agonist application, in order to quantify the time course of desensitisation, or to miniature synaptic currents.

When dealing with fast events, in particular with single channel data, event detection normally is very time consuming because a large number of shut-open and open-shut transitions can occur within a short period of time. Thus 'manual' analysis of a 30 seconds stretch of single channel data may take several hours. The simplest method for detecting transitions in the conductance state relies on the crossing of an (arbitrary) threshold for event detection. A commonly chosen threshold level lies halfway between baseline and single channel amplitude. However, the limited bandwidth of any recording system reduces the apparent amplitude of fast events¹⁵ and leads to problems with their detection, in particular with threshold-crossing detection methods. These problems in single channel analysis are discussed in Chapter 6.

Automated event detection

Most detection methods can be automated (see Sachs, 1983) and the temptation to use them is large because serious single-channel analysis is so time consuming. However, even simple algorithms such as automatic baseline and threshold determination are much less trivial than at first anticipated. The threshold is chosen with respect to a

¹⁵It may be worth remembering at that stage that tape recorders have a limited bandwidth and may be placed in the recording pathway.

reference value (baseline). This baseline is in theory easily determined by the program (for example as mean current in the absence of agonist-elicited single channel openings or whole-cell current). Unfortunately, real baselines often drift because deterioration of the cell-pipette seal causes a slow increase of the holding current. Real baselines may also contain fast artefacts caused by switching on/off of electrical equipment. Thus automatic computer-assisted baseline and threshold settings are much more difficult in practice than in theory. However, the main objection to automated analysis is the fact that the data are not inspected by the experimenter.

Overall, simple detection methods, such as the half amplitude threshold crossing, are useful in circumstances where the data shows good signal-noise ratios, no sub-conductance levels are present and the majority of the single channel openings are long compared to the limits imposed by the bandwidth of the recording system. In contrast, time course fitting (see below), while yielding data of consistently high quality, cannot easily be automated (though a number of operations can be speeded up by careful programming). Whatever method is used the result of this first step in the analysis of single channel data is an idealised trace which consists of a succession of paired amplitude and duration parameters (with or without additional information such as validation parameters etc.) in place of the original data.

Statistical analysis

Statistical analysis of electrophysiological data is concerned with estimating parameters that characterise the idealised data, such as mean and standard deviation of the amplitude and kinetic properties of the event being studied.

Peak or steady state amplitude. The methods used to estimate amplitude are varied and dependent upon the types of data being analysed. Moveable cursors superimposed on a computer-generated display can be used for many forms of data, such as single channel data, with clear defined open and closed levels, or slow events, such as a calcium peak evoked by an agonist.

Other data may require more computationally intensive methods. For example, accurate estimation of the amplitude of small signals, with a poor signal-to-noise ratio, may require averaging of several responses prior to amplitude measurement. When working with single channel data, amplitudes can be estimated, not only by direct measurement, but also by methods which rely on analysis of the distribution of an entire set of data points. For example, calculation of point amplitude histograms (see Chapter 6) saves considerable labour and provides some information about the proportions of time spent at each current level. When dealing with data containing many open levels such analysis may be less useful. In this case more precision may be obtained by calculating the current variance and mean current amplitude for a set of windows of n data points (Patlak, 1988). This window is moved through the data one point at a time, giving values for variance and mean current for n points around each point. The variance is then plotted against mean current at each point. Groups containing a transition between levels have high variance and clusters of points of low variance correspond to current levels at which the trace is stable for some time and thus correspond to discrete current levels.

Time course. As with amplitude measurements, much data can be dealt with by taking measurements with superimposed cursors on a computer-generated display, e.g. the latency of responses to applied agonist. However, in many cases, this method will either be too slow in dealing with large amounts of data, will be unable to produce the information required or will fail to achieve the degree of precision required. In these cases, methods that make greater use of the power of the computer can be used. In the case of synaptic currents, the kinetic information can generally only be obtained by fitting exponential curves to the data (see Chapter 8). In the case of single channel data greater temporal precision can be obtained by fitting the time course of each transition, whereas greater speed but less precision may be achieved with the threshold-crossing method. The relative advantages and disadvantages of these methods are discussed in Chapter 6.

Noise analysis. The unitary events, such as channel gating or receptor potentials, underlying a response occur as random events of characteristic amplitude, lifetime and frequency. The statistical noise they introduce can be analysed to provide information about the amplitude and kinetics of underlying events in signals in which the individual events themselves cannot be resolved (see Chapter 8). Analysis of noise, beyond the most basic studies is impossible without the use of a computer. The required resolution determines the sample rate and length required. The lowest frequency point that can be analysed is given by inverse of the sample length, while the highest possible frequency is half the sample rate to avoid aliasing (Chapter 16). Kinetic analysis of noise requires calculation of the Fourier transform of the signal. This is always done using a Fast Fourier Transform (FFT) algorithm (see Press et al., 1986). Such algorithms impose a limitation upon the way the data is transformed in that they will only operate on arrays whose size is a power of 2, so data can only be processed in blocks of 2^n points. In the past the size of available memory could impose tight limits upon the size of the array that could be transformed. Thus, if each number is represented by a 4 byte floating point number, a sample of 2 seconds at 4069Hz, allowing analysis from 0.5Hz to 2KHz, would take up 32 KB. Using compilers that allow access to the full addressable memory of an 80386 or 80486 processor effectively removes this limitation as arrays much larger than this can be handled.

A second issue concerns the use of integer or floating point numbers to represent the individual data points. Integer FFTs were much faster than floating point FFTs, as CPUs can perform integer arithmetic more rapidly. However, the drawback is that the power spectrum produced, the plot of the signal power vs frequency, is distorted at the bottom end by the limited numerical resolution, the points being clustered at discrete levels. In practice, in a recent computer fitted with a FPU the difference in performance of integer over floating point FFTs is minimal, and not worth the loss of resolution.

In general, analysis of noise requires sampling on two channels, one channel records a low gain, low pass filtered record, for calculation of mean current levels. The second channel records the noise at high gain, after removal of DC components by high pass filtering, and removal of all frequencies above the folding (Nyquist) frequency by low pass filtering. Calculation of spectrum and variance of the noise

record allow information about the amplitude and kinetics of the underlying events to be obtained, as described in Chapter 8.

Curve fitting. The term curve fitting means the process of adjusting the parameters of some equation (e.g. the slope and intercept of a straight line) so as to make the calculated line the best possible fit to the experimental data points. The first thing to decide is what equation is to be fitted and which of its parameters are to be estimated. This, of course, depends entirely on the scientific problem in hand. The second thing to decide is what ‘best fit’ means; in practice, it is usually taken to mean the ‘method of least squares’. Thus, we define a weighted sum of squared deviations:

$$S = \sum_{i=1}^N w_i (y_i - Y_i)^2$$

where y_i is the i th observed value (that at x_i), Y_i is the corresponding value calculated from the fitted equation (using the current values of its parameters), and w_i is the **weight** for the i th observation. Thus S is a measure of the ‘badness of fit’ of the calculated values. The procedure is to adjust the parameters (by means of a minimization algorithm) and so alter the Y_i values so as to make S as small as possible.

The weights are a measure of the precision of the data point and should - if possible - be defined as:

$$w_i = 1/\text{var}(y_i)$$

where $\text{var}(y_i)$ is the square of the standard deviation of y_i . For the purpose of parameter estimation, it is only the **relative** values of the weights that matter. A constant coefficient of variation can, for example, be simulated by taking $w_i = 1/Y_i^2$. If, however, internal estimates of error of the fit are to be calculated, the real standard deviations must be used. For digitized data, it is not usually possible to attach useful weighting to the data points; in this case the weights are ignored (i.e. they are implicitly taken to be the same for each point). In any case, it is generally not possible to calculate the errors of the fit for digitized data, because adjacent points are not independent, so the problem cannot be described by the usual curve fitting theory.

The fitting of distributions to single channel data is a different sort of problem (there is no conventional graph of y against x). It is best dealt with by the method of maximum likelihood (see Chapters 6 and 7).

Minimisation algorithms. Whatever method is used it will always involve changing the values of the parameters so as to minimise some quantity such as S . There are many algorithms available for doing this job, and it is one part of the program that you should not normally need to write. There are two types of minimisation algorithms: the **search** methods (e.g. simplex or patternsearch) and **gradient** types (Gauss-Newton or Levenberg-Marquardt). The search methods tend to be more robust and it is much easier to build constraints (i.e. forcing a parameter to be positive) into the fitting with them, but they are usually slow (especially when they are approaching convergence). Gradient methods tend to require more accurate initial guesses, are difficult to use with constraints, but are often faster. Note that the importance of speed depends on the

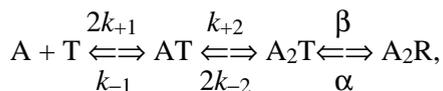
speed itself and on how often you are going to run the program: a 10-fold speed difference may be trivial if it means the difference between taking 1 and 10 seconds to do a fit. It may be rather important if it means the difference between 1000 seconds and 10000 seconds. Most algorithms will find a minimum most of the time: it depends on both the data and the function being fitted. Some data may exhibit local minima when fitted with some functions, in this case no algorithm can be guaranteed to find the smallest of them. However, fitting repeatedly with different initial values may help to obtain a clear estimate of the parameters that give the true minimum.

Understanding the model. The law of computer use ‘garbage in, garbage out’ applies nowhere better than in curve fitting. Each set of data can be fitted by several equations. Thus it is essential to know what the implications of a given equation are. As an example the Hill-Langmuir equation may be used to fit concentration-binding curves and concentration-response curves. The proportion of ligand bound to a population of receptors is

$$p = \frac{x^n}{(x^n + EC_{50}^n)}$$

where x is concentration, EC_{50} concentration for $p = 0.5$ and n estimates the number of molecules of ligand bound per site. In the case of simple binding where the receptor does not change conformation (e.g. open an ion channel) the EC_{50} is an estimate of the dissociation constant (affinity) of binding and n is the number bound per site. However, in the case of concentration-response curves the data may fit perfectly well but the EC_{50} measures an apparent affinity of binding plus conformational equilibrium and n underestimates the number of ligands bound for receptor activation. Thus, there is mechanistic information in the former, but not in the latter case.

We can, for example, use an equation which links the response to agonist concentration to the occupancy of the open state in a four state model in which consecutive binding of two agonist molecules is followed by an additional conformation change to open the channel (see Colquhoun and Hawkes, 1981):



where T represents the unoccupied receptor, A the agonist molecule, AT the singly, A_2T the doubly bound receptor and A_2R the active, conducting, receptor configuration. The transitions between the states are directed by the rate constants k_{+1} , k_{+2} , k_{-1} , k_{-2} , α and β which are the parameters for which we search estimates. In a case such as this it is unlikely that any single experiment will generate enough information to determine all the parameters. In this case one can either (a) determine some parameters from one sort of experiment and then treat these as known values while getting estimates of the others from different experiments, or (b) fit several experiments simultaneously. These parameters are now linked to a mechanism and

may have an actual physical correlate. The equation describing the occupancy of the open state as function of the agonist concentration for the model above is:

$$p_{A2R}(\infty) = \frac{c_1 c_2 \beta / \alpha}{[1 + 2c_1 + c_1 c_2 (1 + \beta / \alpha)]}$$

where

$$c_1 = \frac{x_A}{K_1}, c_2 = \frac{x_A}{K_1}, K_1 = \frac{k_{-1}}{k_{+1}} \text{ and } K_2 = \frac{k_{-2}}{k_{+2}}$$

where x_A is the agonist concentration. The issues involved in composing models to fit data, selecting the most appropriate model and estimating errors are discussed by Colquhoun and Ogden (1988).

Finally, sometimes, one may deliberately fit an equation whose parameters do not have biological correlates in order to get an estimate of a biologically meaningful parameter. For example current-voltage curves may be fitted with a n^{th} degree polynomial in order to get an estimate of the reversal potential V_R for $I=0$.

A large number of different equations may be fitted to both primary (original) as well as secondary data (obtained after analysis). A few examples are listed in Table 2 (for fitting of distributions see Chapter 6).

Many electrophysiology programs contain a curve fitting routine which can handle only a limited number of pre-defined equations. Usually, they cannot handle arbitrary, user defined equations. As a consequence, a good electrophysiology program should include options for exporting all primary and secondary data it produces (see below). A number of fitting programs capable of handling curve fitting with up to 9 and more parameters are now commercially available. Some of them will cope with limited constraints on parameters. Examples include GraphPad, Instat, FigP or Sigmaplot for DOS, Kaleidagraph for the Macintosh. However, many cannot cope with special constraints and in some cases, given time, it may be a good idea to write one's own curve fitting program using published minimising routines. Such subroutines (FORTRAN) have been published by Bevington (1969), Press *et al.* (1986 and references therein) or are commercially available in a compiled form (e.g. NAG library).

Output: exporting the data

Sampled and processed data can be sent to three destinations: screen, printer and data

Table 2. *Some equations used in curve fitting procedures*

Data	Fitted curve
current decay	sum of exponentials
IV curves	n^{th} degree polynomial, Goldman-Hodgkin-Katz
inactivation curve	Boltzman curve
power spectrum	sum of (1/f), Lorentzians
conc-response	Langmuir curve, Hill curve
conc-popen	pA2R (see above)

files. Hardware aspects of graphic displays and printers have been discussed in the first part of this chapter. The necessity to export data to data files arises from at least three reasons: the need for detailed statistics or specialised analysis (e.g. noise analysis), the possibility to fit arbitrary, user-defined curves to the data and finally the necessity to produce publication-quality figures. In the following, we shall discuss various ways to export data and present specific advantages / disadvantages of a few data file formats (see also Table A5).

Text (ASCII) files

The simplest format used to export data consists in transforming all information into user-readable text using the ASCII code. This format is universally readable even across various computer types (DOS, OS/2, Macintosh OS). Most if not all commercially available programs (spreadsheets, statistics, graphics) can read ASCII files. ASCII export / import routines are easily implemented in user-written programs. Finally, ASCII files can easily be read or modified using simple text editors (DOS edit, Windows Notepad, etc.). ASCII files are unfortunately quite uneconomic in terms of storage space: Numbers are no longer encoded in a fixed number of bytes but require a byte per digit. It is worth checking how the various programs interpret ASCII files since some programs (e.g. Lotus 1-2-3) cannot directly handle ASCII files containing both text and numbers. Additional data parsing may be required to restore the original appearance of the data. Other programs (e.g. Kaleidagraph) cannot contain text and numbers within the same column. If the exported data consists of several columns of numbers or text, they must be separated by a character that is recognised as such by the reading program. Commonly used separators are space, comma and tab characters.

Lotus (.WK%) file format*

This format is a custom binary file format used by Lotus Development to encode data within its Lotus 1-2-3 spreadsheet program. Lotus 1-2-3 is a widespread program which runs on multiple platforms (DOS, Windows, OS/2, Macintosh). Furthermore it can be read by several other spreadsheet programs such as MS-Excel or Borland Quattro Pro. Several graphics programs (Lotus Freelance, Harvard Graphics) also can read Lotus 1-2-3 (*.WK%) files. These files can contain text, numbers and graphics information for plots. The disadvantage is that it is not as universally recognised as ASCII files. Determined programmers can incorporate code to read and write Lotus 1-2-3 files into their programs (see Lotus reference).

Proprietary file formats

A recent and promising development is the use of standardised file formats for digitised signals by commercially available software. Both CED and AXON use structured binary data files so the user may extract whatever information he or she requires. CED (and others) have gone a step further by writing and distributing a set of subroutines written in Pascal or C which can read and write standard CED Filing System ('CFS') files (and other files using proprietary file formats). An immediate advantage of standardisation is that data can be easily exchanged between programs

and users. Thus data sampled with CED's CHART program can be analysed with PATCH or Spike2. Similarly, a number of researcher-written programs for single channel analysis, such as IPROC and LPROC, can read the Axon pCLAMP data file structure (ABF: Axon Binary File). Unfortunately, the ABF and CFS file structures are quite different. Programs designed to handle CFS files cannot directly read ABF files, (a file conversion program ABF to CFS is shipped with CED's EPC suite) and vice versa. So far, no single standardised data file structure has yet been widely accepted.

Graphics file formats

The two most important types of graphics files are bit-mapped (raster) and vector-graphics (object-oriented) files. The main difference between those two file types is that vector-based files can be enlarged without loss of precision.

Bit-mapped. Common bit-mapped file formats are TIFF (Tag Image File Format, codeveloped by Aldus) and PAINT files (common on Macintoshes). Raster files are a little like photocopies or photographs. They can be modified but usually not at the level of individual elements of the picture.

Vector-graphics. Vector-based graphics files can more easily be scaled and modified. PostScript files are text (ASCII) files which describe a vector image. Encapsulated PostScript Files (EPS Files) include a PostScript text file and an optional low resolution preview of the underlying image. The data file can be written as a text (ASCII) or binary file. Interestingly, EPS files can also be used to save raster images though they do not perform a raster to vector transformation. The HPGL graphic file format is also a vector-based graphic file format which enjoys a certain amount of universality (see hardware section) and is used by certain electrophysiology programs to export graphs (see Table A2).

Metafiles. This graphic files format contains both raster and vector graphics. Such files are PICT files on Macs and Windows Metafile Format (WMF) on PCs. They are the Clipboard standard files. Another metafile format is the CGM (Computer Graphics Metafile) format, a format defined both by the International Standards Organisation (ISO) and the American National Standard Institute (ANSI).

There is a certain amount of compatibility and exchangeability in that some graphic file formats are shared by several programs. This is true of Metafiles within DOS (CGM) and Windows (WMF). HPGL (and Lotus PIC) files are recognised by a number of DOS and Windows programs too. If export across platforms (DOS /Windows to Mac and vice-versa) is required the best bet may be to use TIFF and EPS graphic files. No rule can be made though about which file types can be exported or imported by any given program (see Sosinsky, 1994).

8. Writing your own program

Thus far, the discussion has concentrated upon the nature of a good program and the extent to which those currently available meet the criteria. Although these programs cover most of the common forms of electrophysiological experimentation, there can

be no doubt that occasions still exist where it is necessary for the experimenter to write his/her own program.

One language does not fit all

Computer programming can be done at three levels: Low level language (Assembler) programming is for the experts. Assembler reaches parts of the computer other languages cannot reach. It is hardest to write, but fastest in execution time. At a moderate level of programming expertise, 'high level' programming languages (BASIC, FORTRAN, Pascal, C, Modula etc.) enable all (arithmetic) data manipulation usually required in a lab. Solving given mathematical problems requires for example, understanding of the mathematical / statistical background behind the data manipulation. The reader should be aware that knowing that solving a given problem requires inverting a matrix is only the first part of the solution of this problem. It is usually just as important to have access to specialised literature on so-called 'numerical recipes' (e.g. Press *et al.*, 1986) to perform given mathematical tasks (e.g. matrix inversion). Specialised applications such as writing software to convert binary file formats require additional knowledge on proprietary file formats which is usually provided by the software companies (as manuals) or published (e.g. Lotus File Formats). Most commercially available programs were written in one these high level languages. At the highest (language) level, some commercial programs enable the user to simplify repetitive tasks in data manipulation so they can run in a predetermined, 'programmed' way, using so-called macros (e.g. Lotus 1-2-3). Recently, a number of programs have appeared which can perform analytical mathematics (as opposed to numerical). Such programs include Wolfram Research's *Mathematica*, MathSoft's *Mathcad* or Waterloo Maple Software's *Maple*.

It is striking how many of the existing commercial programs originated in research laboratories (e.g. Axon's pCLAMP and Axograph, Instrutech PULSE software or CED PATCH & VCLAMP modules). A case can therefore be made for the need to retain programming skills within at least some electrophysiological laboratories. New forms of signals such as those obtained from fluorescent dyes often require the recording of signals from new classes of devices (e.g. photomultipliers or video cameras) and different analysis algorithms.

A number of programming languages are available for both the DOS and Macintosh platforms. They include numerous brands of BASIC, Pascal, FORTRAN, Modula 2, Assembler and C, to name but the most important. Programs in a high level language can be pepped up with low-level subroutines: this so called mixed language programming may be an excellent approach for solving certain problems. A program written in a 'high level' language makes use of compiled subroutines written in 'lower level' languages (Assembler, C) which run very fast. This removes the task of writing and checking specialised subroutines from the occasional programmer while giving him/her access to high powered software. The reader is referred to specialised literature for this sort of programming (e.g. Microsoft mixed language programming guide).

One critical application of this approach to anyone writing data acquisition or

experimental control software is the provision of libraries of low level procedures that handle communication with the interface. Programming these aspects generally involves lengthy Assembler programming. Most laboratory interfaces are supplied with libraries of user callable procedures (e.g. Axon's set of subroutines for the Digidata 1200 or the LabPac set of subroutines for the Labmaster DMA card), or other support mechanisms, that handle those low level aspects. When selecting an interface with the intention of writing one's own software, the quality and functionality of the programmer support is a major factor to consider. Other examples of user callable subroutine packages (libraries) are: the CFS set of subroutines to access CFS files, the scientific NAG (National Algorithms Group) library (state of the art statistics and mathematics package), or the MetaWindows set of graphic subroutines.

The widespread adoption of GUI based systems, such as Microsoft Windows and the Macintosh OS calls for programming languages adapted to these new environments. Several programming languages are available for the Windows platform and the Macintosh OS. Microsoft's Visual Basic, Borland's Turbo Pascal for Windows (and other similar tools) are fairly easily learnt languages which enable writing of true Windows applications (if somewhat slower than applications written in C). These programming environments, together with the latest versions of Microsoft and Borland C++, all support a visual programming metaphor for the construction of the Windows user interface. This removes from the programmer a major part of the load of developing a Windows application, allowing the programmer to concentrate on coding the underlying functionality in a traditional programming language. Mixed language programming with all its advantages is possible (also from Visual Basic) by calling so-called direct link libraries (DLLs) which perform specialised tasks (e.g. data acquisition or display). Some of the laboratory interfaces presented in Table A1 are supported by DLLs. However, as discussed earlier, the development of such programs is more difficult than writing straight DOS programs. This may become unavoidable when multi-tasking systems such as Windows NT become standard. Using the Macintosh toolbox is already now a difficult task for non-professional programmers. Another approach may be to use DOS compilers which take advantage of the protected (true 32 bit) processor mode. With the exception of the latest generation of C++ programming environments, these may have the disadvantage of not being mainstream.

A further solution to the problem of developing customised laboratory software within the new GUI environments is to make use of a specialised software development system which provides the basic software components necessary for the acquisition, display and analysis of analogue signals. One of the most advanced examples of this approach is the National Instruments LabView package which allows the development of data acquisition software for both Windows and the Macintosh. With this package, 'virtual instruments' can be created by graphically 'wiring up' acquisition, analysis and display icons on the screen. National Instruments also produce the Lab-Windows package which allows the development of similar systems (for PCs only), but with a more conventional programming approach, using enhanced versions of the C and QuickBasic programming languages.

In each case, the merits of developing a new program in-house should be carefully

weighed, with consideration given to whether there might be better uses for one's time. Software development, even with the assistance of a good development environment, is a skilled and time consuming activity. There is little value in producing a poor and unreliable imitation of an existing program. On the other hand, the existing software that is widely available is far from perfect, and there is always a place for faster, more versatile, easier to use, software.

9. Concluding remarks

The preceding sections are intended to provide sufficient basic information and guidance to allow a suitable laboratory computer system to be selected. However, it is without doubt a good idea to take additional advice, where possible, from colleagues in your own field who have experience of performing experiments and analysis similar to those that you intend to undertake, with one or more of the packages under consideration. In addition, it is crucial that before making a selection, a specification of requirements should be drawn up which details what functionality is needed such as the number of signals to be recorded in parallel, maximum sample rates (which will be determined by signal bandwidth), signal length, short and long term storage requirements (which will be related to the sample rate and length), provision of control signals and data or trigger outputs to monitor scopes, forms of analysis required, forms of hard copy needed. Other factors such as cost limitations or constraints imposed by existing hardware must also be considered.

The field of personal computers is moving fast. It is quite obvious that by the time this chapter is published or soon after, new, better and faster processors will be available. This may be important to satisfy the increasing demands of number crunching applications and of graphical user interfaces. The ease of use and affordability of personal computers has created a trend for people to buy ready to use 'turn-key' systems. This is a clear change from the time of mini-computers such as the PDP-11 when all software had to be custom made. There is a danger that electrophysiologists do only those experiments that can be done with commercial software instead of doing those experiments necessary to answer a (biological) question. The clear trend towards GUIs may aggravate this situation by further decreasing the number of people both willing and able¹⁶ to write their own software.

This means that electrophysiologists have to rely on programs written by people who are not necessarily familiar with the biology behind the numbers. Electrophysiologists using such programs should be aware of this fact and critical about what computers tell them. To those who have not got access to the program source, the hardware and software should be treated as a black-box whose behaviour should be carefully assessed and critically analysed to be sure its limitations are appreciated. Numbers need not be correct just because they appear on a terminal or a computer print out. Users should never forget that while computers are infinitely

¹⁶ The limits often being set by time, budget and publishing considerations.

diligent they are also infinitely dumb. By taking away the tedious task of number 'crunching' computers have if anything increased the responsibility of users to know and understand the theories and models underlying their calculations.

Acknowledgements

Several discussions with David Colquhoun, David Ogden, Jörg Stucki, as well as with Peter Rice and Greg Smith, using various means of communication (phone, fax and electronic mail) have greatly enhanced the accuracy of this chapter. People often identify strongly with their computers: the opinions expressed here are those of the authors or a hard fought compromise thereof.

References

Original articles and books

- ARMSTRONG, C. M. & BEZANILLA, F. (1974). Charge movement associated with the opening and closing of the activation gates of the Na channels. *J. Gen. Physiol.* **63**, 533-552.
- BENDAT, J. S. & PIERSOL, A. G. (1986). Random data: analysis and measurement procedures. New York: Wiley Interscience.
- BEVINGTON, P. R. (1969). Data reduction and error analysis for the physical sciences. New York: McGraw Hill.
- COLQUHOUN, D. & HAWKES, A. (1981). On the stochastic properties of single ion channels. *Proc. R. Soc. Lond. B.* **211**, 205-235.
- COLQUHOUN, D. & SIGWORTH, F. J. (1983). Fitting and statistical analysis of single-channel records (Ch.11) In *Single-Channel Recording*, (ed. B. Sakmann and E. Neher). New York, London: Plenum Press.
- COLQUHOUN, D. & OGDEN, D. (1988). Activation of ion channels in the frog end-plate by high concentrations of acetylcholine. *J. Physiol. (Lond)*, **395**, 131-159.
- DEMPSTER, J. (1993). Computer analysis of electrophysiological signals. London: Academic Press.
- LOTUS FILE FORMATS FOR 1-2-3 SYMPHONY & JAZZ. File structure descriptions for Developers. Lotus Books. Reading, MA, USA: Addison-Wesley Publishing Company.
- PATLAK, J. B. (1988). Sodium channel subconductance levels measured with a new variance-mean analysis. *J. Gen. Physiol.* **92**, 413-430.
- PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., & VETTERLING, W. T. (1986). *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press.
- SACHS, F. (1983). Automated analysis of single-channel records. (Ch. 12) In *Single-Channel Recording*. (ed. B. Sakmann and E. Neher). New York: Plenum Press.
- SOSINSKY, B. (1994). Graphically speaking. *MacUser* Jan. 1994. pp.145-149.

Useful articles in BYTE

Although BYTE is not the only source of information on computers it was chosen because it has become an authority in the field and is likely to be available in libraries.

- FPU Face-off. *BYTE* Vol **15** (12), 194-20. (Nov. 90)
- Managing gigabytes. *BYTE* Vol **16** (5), 150-215 (May 91).
- Operating systems trends. *BYTE* Vol **17** (10), 158-197 (Oct. 92).
- PowerPC performs for less. *BYTE* Vol **18** (9), 56-78 (Aug. 93).
- Printers face off. *BYTE* Vol **18** (12), 276-307. (Nov. 93).

State of the art mass storage: magnetic, optical, holographic and more. *BYTE* Vol 15 (12), 272-341 (Nov. 90).

Windows accelerators. *BYTE* Vol 18 (1), 202-206 (Jan. 93).

Suppliers

Axon Instruments Inc., 1101 Chess Drive, Foster City, CA 94404, USA

T: +1 415 571 9400, F: +1 415 571 9500

Cambridge Electronic Design Ltd, The Science Park, Milton Road, Cambridge, CB4 4FE, UK.

T: +44 223 420186, F: +44 223 420 488

Data Translation, 100 Locke Drive, Marlboro, MA 01752, USA

T: +1 508 481 3700, +1 508 481 8620

Dagan Corporation, 2855 Park Avenue, Minneapolis, MN 55407, USA

T: +1 612 827 6535

Digitimer Ltd, 37 Hydeway, Welwyn Garden City, Herts., AL7 3BE, UK

T: +44 707 328347.

Frequency Devices Inc., 25 Locust Street, Haverhill, MA 01830, USA

T: +1 508 374-0761, F: +1 508 521 1839

Fylde Electronic Laboratories Ltd, 49/51 Fylde Road, Preston, PR1 2XQ, UK

T: +44 772 57560.

Instrutech Corp., 475 Northern Blvd, Suite 31, Great Neck, NY 11021, USA

T: +1 516 829 5942, F: +1 516 829 0934

NAG Ltd, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR, UK

T: +44 865 511 245, F: +44 865 310 139

National Instruments, 6504 Bridge Point Parkway, Austin, TX 78730-5039, USA

T: +1 512 794 0100, F: +1 512 794 8411

Scientific Solutions, 6225 Cochran Road, Solon, OHIO 44139, USA

T: +1 216 349 4030, F: +1 216 349 0851

Appendix

The tables in the following appendix summarise typical commercially available hardware and software solutions available for electrophysiological (and other lab) data acquisition and analysis, as of January 1994. The data presented were either obtained directly from the manufacturers, or taken from their literature. As with all issues relating to computer hardware and software, these systems and packages are subject to rapid change. Therefore, this appendix is intended to provide a guide to the type of solutions that may be available, and the data provided should be checked with the manufacturers before any purchasing decisions are made.

Manufacturer's addresses are listed at end of chapter. All boards are supported by various software classified as bundled (B) with the product, additional (A), i.e. available at additional cost and third party (T).

Table A1. Comparison of commonly used laboratory interfaces

Interface	CED 1401 <i>plus</i>	Digidata 1200	ITC-16	DT 2839	Labmaster AD	AT/NB ¹ -MIO-16
Supported by electrophysiology software	yes	yes	yes	no ²	no ³	no
Computer type	PC, PS/2, MacIntosh	PC	PC, MacIntosh, Atari	PC	PC	PC, PS/2, MacIntosh
Analog input(s)	16 SE ⁴ 32 option	32 pseudo DI	8 DI ⁵	16 expandable to 224	16 SE / 8 DI expandable to 64 se or 32 di	16 SE / 8 DI
ADC range (s)	±5 V standard ±10 V option	±10.24 V	±10.24 V	0-1.25, 0-2.5, 0-5, 0-10 V, ±1.25, ±2.5, ±5, ±10 V	±10 V	0-10 V, ±5 V, ±10 V
AD converter	12 bit 333 kHz	12 bit 333 kHz	16 bit 100 kHz / 12 bit 200 kHz	12 bit 224 kHz (1 MHz 1 ch)	12 bit 330 kHz	12 bit 200 kHz ⁶
Programmable input gain	option ⁷	1, 2, 4, 8	no	1, 2, 4, 8	PGH: 1,2,4,8 or PGL: 1, 10, 100,1000	0.5, 1, 2, 5, 10, 20, 50,100
DA converters	4× 12 bit 444 kHz	2× 12 bit 333 kHz	4 × 16 bit 200 kHz	2 × 12 bit 130 kHz; 260 kHz aggregate	2× 12 bit 130 kHz full scale max 500 kHz	2× 12 bit 250 kHz ⁸
DAC range(s)	±5 V standard ±10 V option	±10.24 V	±10.24 V	0-5 V, 0-10 V, ±2.5 V, ±5 V, ±10 V	±10 V (5 mA)	0-10 V, ±10 V
Asynchronous digital input(s)	16× TTL	8× TTL	16× TTL	16× TTL	8× TTL	8× TTL
Asynchronous digital output(s)	16× 24 mA	8× 24 mA (or 4 sync.)	16× 24 mA	16× 24 mA	16× 24 mA	8× 24 mA
Clock(s)	4× 32 bit 250 ns 1× 48 bit 1 μs	48 bit 250 ns	250 ns	2× 16 bit 250 ns programmable	16 bit 250 ns	16 bit 200 ns programmable
Counter/timer	2 channels	9513A 5 channels	no	9513A 5 channels	9513A 5 channels	9513A 5 channels

¹AT indicates PC cards, NB (NuBus) Mac cards. ²Older board (DT2801A) is supported by Strathclyde software and other packages. ³Older board (Labmaster DMA) is supported by Strathclyde software and other packages. ⁴SE: single-ended, DI: differential. See section 4 for definition. ⁵Optically isolated. ⁶NB-MIO: 100 kHz. ⁷Gains: 1, 2, 5, 10, 20, 50, 100. ⁸Option 64 F5: 1 MHz.

CED 1401_{plus} (CED Ltd, UK)*Hardware*

Connection to computer via 1 short ISA or MCA card for PCs, 1 NuBus card for Macs. Stand alone operation possible thanks to 32 bit 10 MIPS built-in microprocessor. Up to 17 MB on-board memory. All input/outputs (ADC, DAC, Clock, Digital I/O) are brought to BNC and D type connectors on front panel of a 19 inch box for connection to lab equipment.

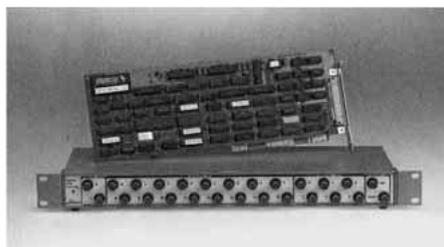
*Software*

B: utility programs with source in C, QBASIC, PASCAL, commands for internal processor

A: Spike2, Chart, Patch, Signal Averager, EEG, ECG, FFTs, Windows 3.1 and Windows NT DLLs, device drivers and support. C support and device driver support for Macintosh.

Digidata 1200 (Axon Instruments, USA)*Hardware*

Consists of 1 ISA bus card. Connection to lab equipment via optional: BNC interface, screw panel terminal and Cyberamp adapter(s).

*Software*

B: utility programs, support library (+ source in C and BASIC), Windows DLL in development

A: pCLAMP (patch clamping / microelectrode recording), AxoTape (strip chart recorder with real time display), AxoBasic (development environment for creating acquisition and analysis programs).

ITC-16 (Instrutech, USA)*Hardware*

Connection to computer via 1 NuBus (MacIntosh) or 1 ISA card (PCs). Partial stand-alone operation via local microprocessor. Connection to lab equipment via rack mount case with BNC connectors.

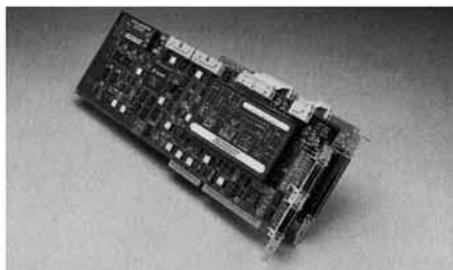
Software

B: IGOR Xops, QBasic drivers, C Drivers

A: PULSE (Heka), AXODATA (Axon), SYNAPSE, Pulse Control.

DT2839 (Data Translation, USA)*Hardware*

Consists of 1 ISA/EISA bus card. Connection to lab equipment via simple screw terminal or expander board.

*Software*

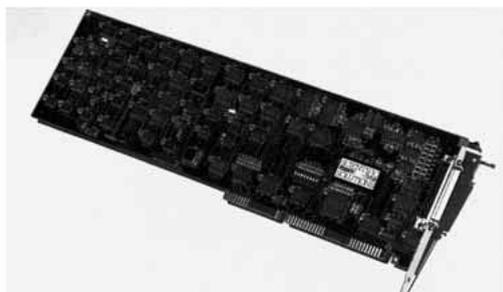
B: Gallery, example programs, Windows 3.1 DLL, C software developer's kit for Windows 3.1, Toolkit subroutine libraries

A: Visual Basic Programming tools, custom controls for data acquisition and high speed plotting Labtech Notebook, Control

T: P-DAAS (Pentech), QEDIESing 1000 (Momentum), Experimenter's Workbench/Discovery(Brainwaves).

Labmaster AD (Scientific Solutions, Inc, USA)*Hardware*

Consists of 1 16 bit ISA (EISA compatible) card. Connection to lab equipment via 'LabRack' optional BNC termination panel.



Software

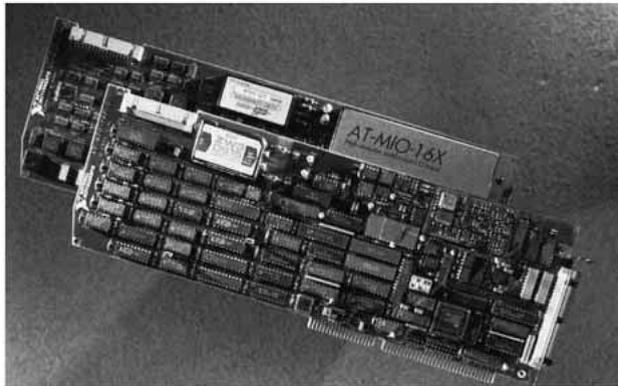
B: utility and test software, software library for most MS DOS languages, data streaming oscilloscope software

A: paperless strip chart recording, data logging, storage scope emulation

T: Snapshot, Notebook/Control, DADisp, EP/EEG Scan, DriverLINX, RT-DAS, Data Sponge, ASYST. Supported by Software Toolkit (not by LabPac).

AT/NB MIO-16*Hardware*

Consists of 1 ISA/MCA (PCs) or 1 NuBus (MacIntosh) card. Connection to lab equipment via optional screw terminal (BNC) or conditioning modules.

*Software*

B: DAQWare (integrated package), NI-DAQ for DOS (most MS DOS languages), NI-DAQ for Windows (DLLs). Library of functions & utilities for Macintosh OS

A: Labwindows for DOS, LabVIEW for Windows/MacIntosh/Sun, LabTech Notebook, HT Basic. Measure (Add on for Lotus 1-2-3).

Table A2. *Hardware requirements of some commercially available electrophysiology packages*

Author/manufacture	Axon Instruments	CED Ltd	HEKA Elektronik	John Dempster
Name/Version	pCLAMP 6.0 June 1993	PATCH+VCLAMP 6.03 June 1993	PULSE + PULSEFIT 7.0 October 1992	STRATHCLYDE EP SUITE
Source code	MS Basic PDS 7.1, Borland Pascal, 7.0 some routines in MS C / Assembler	Borland Pascal 7.0 + some Assembler	Modula-2	Microsoft FORTRAN & Assembler
Source code available	Source code for analysis provided free of charge with package, source code for acquisition not available	enquire	yes, non disclosure agreement required	yes
Approx price	US\$ 2000.	UK£ 1000.	US\$ 1500.-	Free to academics (shareware)
Site licence available (multi-user)	by department	yes	no	no
1. Computer equipment				
Operating system / Version	DOS-3.1 , DOS 5.0 or higher recommended	DOS 2.0 or higher	Apple OS 6.03 or higher	DOS 3.3 or higher
Computer type / (Processor)	AT 33 MHz 80386 DX or higher recommended	PC/AT 80×86	Macintosh II, Quadra	PC/AT 80×86
Maths co-processor required	required	recommended	required	recommended
RAM minimum / recommended	640 kB (+ 1 MB disk cache recommended)	640 kB / 1 MB	8 MB	512 kbyte/640 kbyte
Diskspace minimum	7 MB for pCLAMP files	1.5 MB	3 MB	2 MB
Max screen resolution	VGA (640×480)	VGA 640×480 Wyse 1280×800	unlimited	640×480
ADC interface(s) supported	Axon Instruments Digidata 1200, Scientific Soltns Labmaster DMA or MC/DAS (PS/2)	CED1401 / 1401 <i>plus</i>	Instrutech ITC16	CED 1401/1401 <i>plus</i> , Data Translation DT2801A, DT2821, DT2812, Scientific Solutions Labmaster, National Instruments LAB-PC, ATMIO- 16X.
2. Printers/plotters				
Plotters	HPGL (≤ 6 pens)	HPGL	yes ¹	
Dot matrix printer	many types	Epson or -compatible	yes ¹	any Epson compatible
Inkjet printer	DeskJet	DeskJet	yes ¹	DeskJet
Laser printer	many types incl. HP LaserJet	HP-LaserJet or compatible	yes ¹	HP LaserJet or compatibles
PostScript laser printer	yes	no	yes ¹	yes
3. Electrophysiology equipment				
Recommended amplifier(s)	any	any	EPC-9	any
4. Computer controlled lab equipment				
Amplifier / signal conditioner	CyberAmp ²	no ³	no	
¹ any Macintosh OS compatible.				
² CyberAmp: Axon Instruments' signal conditioner. See text.				
³ CED1902 signal conditioner and Axon CyberAmp supported by programs SIGAV, CHART and SPIKE2.				

Table A3. Comparison of 4 commercially available electrophysiology programs.
Data Acquisition characteristics

	pCLAMP	PATCH + VCLAMP	PULSE + PULSEFIT	STRATHCLYDE SUITE ¹
1. General				
Min interval	3-10 μsec^2	3 μsec	5 μsec	10 μsec^2
Max interval	1 sec	1 sec	65 msec	15 msec-10 sec ²
Split clock sampling	yes	no	no	no
Input channels	8	rep.samp.: 4 cont.samp.: 1	out:2, in:2	6
Amplifier monitoring	yes	yes	yes	no
Filtering during acquisition	no	no	yes	no
2. Repetitive sampling				
(a) acquisition				
Exact episode timing	yes	yes	yes	
Linking of stimulus families	yes	yes	yes	
Seal test	automatic	standard pulse	standard pulse	standard pulse
Max number of points/pulse	16 k samples	10 k samples	16 k samples	2 k per channel
Adjustable trigger pulse	yes	yes	yes	yes
On-line leak subtraction	yes	yes	yes	no ³
Leak subtraction method	P/N ⁴	P/N	P/N	P/N
On-line averaging	yes	yes	yes	yes (1 channel)
(b) Command pulses				
Command types	steps/ramps ≤ 8 transitions	steps/ramps ≤ 2 transitions	steps/ramps/ sine waves 16 k transitions	steps/ramps ≤ 3 transitions
Free definable command	up to 16 k points memory/file	≤ 100 dig pulses on 8 ch / sweep	file	
Auto increment/decrement	amplitude +duration	amplitude + duration	amplitude + duration	amplitude + duration
Arbitrary order of V steps	user-defined list	on line user control	interleaved alternate	interleaved alternate
Conditioning pulses	yes	yes	yes	yes
3. Continuous sampling				
Max rate to disk	80 kHz ⁵	80 kHz ²	200 kHz ²	30 kHz (PAT)
Max data file size	4 GB	4 GB	2 GB	32 MB
Watch - write mode switch	yes	yes	yes	yes
Event-triggered acquisition	yes	yes	yes	yes
Adjustable pre-trigger	yes	no ⁶	no	yes
Markers during sampling	no	yes	no	no

¹Consists of following programs: WCP: whole cell analysis, PAT: single channel analysis, SCAN: synaptic current analysis, VCAN: voltage clamp analysis and SPAN: power spectrum/variance analysis.

²Depends on interface used.

³Implemented off-line.

⁴P/N: Pulse amplitude P is divided by arbitrary N, applied N times and scaled by N to obtain leak signal.

⁵Up to 333 kHz using appropriate hardware.

⁶Achievable by adjusting data block size.

Table A4. Comparison of 4 commercially available electrophysiology programs.
Data Analysis characteristics

	pCLAMP	PATCH + VCLAMP	PULSE + PULSEFIT	STRATHCLYDE SUITE
1. General				
On-line analysis	yes ¹	yes ¹	yes ²	no
Mathematical operations between episodes	yes	no	yes	no
2. Amplitude domain				
Point histogram (max bins)	max 256 bins	max 1024 bins	yes	max 512 bins
Probability density- amplitude histogram	yes	yes	yes	yes
Amplitude vs time histogram	yes	yes	yes ³	no
Automatic peak - trough follow	yes	yes	yes ³	?
3. Time domain				
Open time histogram	yes	yes	yes ³	yes
Shut time histogram	yes	yes	yes ³	yes
Sigworth Sine histogram	max 512 bins	no	yes	see below
Rise time to peak in whole cell current	10-90% rise time	yes	yes	10-90% rise time
First latency of single channel data	yes	yes	yes	no
4. Curve fitting				
Search / optimizing algorithm	various ⁴	Simplex	Simplex	Marquardt-Lev.
Gauss to point histogram	yes up to 6	yes up to 6	yes	yes
Exp. to dwell times	yes up to 6	yes up to 2	yes	yes (PAT)
Exp. to current response	yes up to 6	yes	yes	yes (WCP)
Other curves	yes ⁵	yes ⁶	yes ⁷	yes ⁸
5. Single channel analysis/detection method/burst analysis				
Threshold crossing	yes	yes	yes	yes ⁹
Automated analysis	yes	yes	yes	yes
Idealized trace	yes	yes	yes	yes
T-crit adjustable		yes	yes	yes
Burst length	yes	yes	yes	yes
Closings per burst	yes	yes (openings)	yes	yes
Burst count	yes	yes	yes	yes
6. Power spectrum/noise analysis				
Available	no	no ¹⁰	yes	yes

¹Partial; ²All aspects except curve-fitting possible; ³Program Mac TAC reads pCLAMP (ABF) and PULSE files; ⁴Marquardt-Levenberg, LSQ, Simplex (SSQ/max likelihood), Chebychev; ⁵Boltzman, power, linear regression, exponential with sloping baseline; ⁶Geometrics for burst analysis; ⁷Polynomial up to 9th degree, double exp., Hodgkin & Huxley, Boltzman, Linear (IV)GHK, Activation, high voltage block, low-voltage block; ⁸Hodgkin & Huxley; ⁹Uses this method only to compute an amplitude histogram; ¹⁰Spectral analysis functions in Spike2 and Waterfall (window definable).

Table A5. Comparison of 4 commercially available electrophysiology programs.
Data handling characteristics

	pCLAMP	PATCH + VCLAMP	PULSE + PULSEFIT	STRATHCLYDE SUITE
DATA FILE INFO				
Session logging/ protocol file	user actions recorded in log file with time stamping	short table in data file	yes, incl. text editor	time stamped session log program WCP
Automatic filename implementation	yes	yes	no	no
Notepad in data file	yes	yes	yes	yes
DATA IMPORT / EXPORT				
1. Data file format				
Proprietary file format	yes ¹	yes ²	yes	yes
Documentation	yes	CFS User guide	Pulse User guide	yes
Dedicated user software	yes ³	yes ⁴	yes ⁵	no ⁶
Export formats	ABF, ASCII	ASCII, CFS	ASCII, IGOR text/binary, spreadsheet, PICT	ABF, ASCII, CFS
Import formats from	ABF, ASCII	ABF, CFS	Atari	ABF, ASCII, CFS
2. Handling of results of analysis				
Idealised trace editable	yes	yes	yes	yes
Histogram data	yes	yes	yes	yes
IV data	yes	yes	yes	yes
Other data		yes ⁸	yes ⁷	
3. Available XY Plots				
Choice of X	yes ⁹	yes ¹⁰	yes ¹¹	yes ¹²
Choice of Y	yes ⁹	yes ¹⁰	yes ¹³	yes ¹²
4. Graphics				
Export file format	HPGL	HPGL	PICT File max32kB	HPGL
Screen dump	to printer	to printer	no ¹⁴	no

¹ABF (Axon Binary File), see text

²CFS (CED Filing system), see text

³Source in QBasic, C and PASCAL included

⁴Libraries for MS-DOS languages as well as WINDOWS DLL

⁵Into IGOR, ASCII spreadsheet, sample code for user programmable exports

⁶Program source code available

⁷Any parameter (experimental or computed) can be plotted vs most others and fitted

⁸Raw ADC, curve fit param.

⁹Results of analyses, ADC channels, data files

¹⁰I and V differences, index, time, peak/trough value, user defined

¹¹Voltage, duration, real time, index, user defined

¹²Peak, average, rise time, rate of rise, variance, inter-record interval

¹³Extreme, min, max, mean, var, charge, Cslow, Gseries, user defined

¹⁴Requires utility program