Chapter 3

Guide to Data Acquisition and Analysis

STEFAN H. HEINEMANN

1. Introduction

This chapter should provide a first guide to the acquisition and analysis of patch-clamp data. Except for the sections on single-channel analysis, the procedures and considerations also hold for data obtained using other voltage-clamp methods. It is assumed that the reader is familiar with the standard methods and terminology of patch-clamp electrophysiology. Because many of the problems that arise during data analysis can be avoided by a proper design of the experiment, including data acquisition, we start by deriving some criteria that should be considered before actually starting to record data.

Performing electrophysiological experiments and, in particular, analyzing them are tasks made easier using a high degree of automation, which can be provided by an increasing variety of computer hardware and software. This means that, except for very simple applications, one has to decide what kind of computer system, including peripherals and software packages, should be used for the experiments. Because of the rapid turnover of hardware and software products it is impossible to provide a complete overview of the available components. Therefore a continually updated list of products, specifications, and vendors is deposited on a public-domain data base. Access to this data base is described in Chapter 1 of this volume. More information on specific products and analysis methods can be found in French and Wonderlin (1992), Dempster (1993), and in brochures and reference manuals such as *The Axon Guide* (Axon Instruments) from various vendors of patch-clamp hardware and software. The major aim of this chapter is to derive criteria for the development or purchase of acquisition and analysis software.

The next section illustrates what kinds of analysis tasks exist and how they can be accomplished with various software configurations. In particular, it is important to consider which features are essential for a successful experiment and whether these features should be supplied by dedicated acquisition and analysis programs or by general-purpose programs.

After a brief introduction to what should be considered during the acquisition of current data (Section 2), an overview of analysis procedures applied to single-channel and macroscopic current data is given. Only the basic principles, advantages, and disadvantages are discussed. For more detailed theoretical treatments and for numerical implementations the reader should refer to the cited literature and to the theoretical analysis chapters later in this volume.

STEFAN H. HEINEMANN • Max Planck Society, Research Unit, "Molecular and Cellular Biophysics," D-07747, Jena, Germany.

Single-Channel Recording, Second Edition, edited by Bert Sakmann and Erwin Neher. Plenum Press, New York, 1995.

Section 6 is a buyers' guide for computer hardware and software. The intimate connection between hardware and software, which poses considerable restrictions once decided on, is discussed. In particular, an introduction is given to the different approaches to solving software problems. These range from the purchase of ready-to-go programs for electrophysiology to the development of user software from scratch.

1.1. Levels of Analysis

The ultimate aim of the digital acquisition of electrophysiological data is to allow both quantitative and qualitative analysis of the biological system under consideration. Evaluating the change in shape of compound action potentials on alteration of experimental conditions is an example of qualitative analysis, whereas the determination of the equilibrium binding constant of a molecule to an ion channel based on the measurement of single-channel kinetics is highly quantitative. Although for the first example only data recording, timing, and display features are required, the second example demands more dedicated single-channel analysis functions.

In Fig. 1 an outline of various levels of analysis is given. This starts with the conditioning and acquisition of the data and is followed by display and preanalysis. The major part comprises analysis dedicated to the very specific problems arising in electrophysiology at the level of raw data or on parameters that were derived at earlier stages of analysis. At all levels analysis may have become so generalized that the use of multipurpose graphics or spreadsheet programs is possible. Alternatively, very specialized analysis tasks may arise

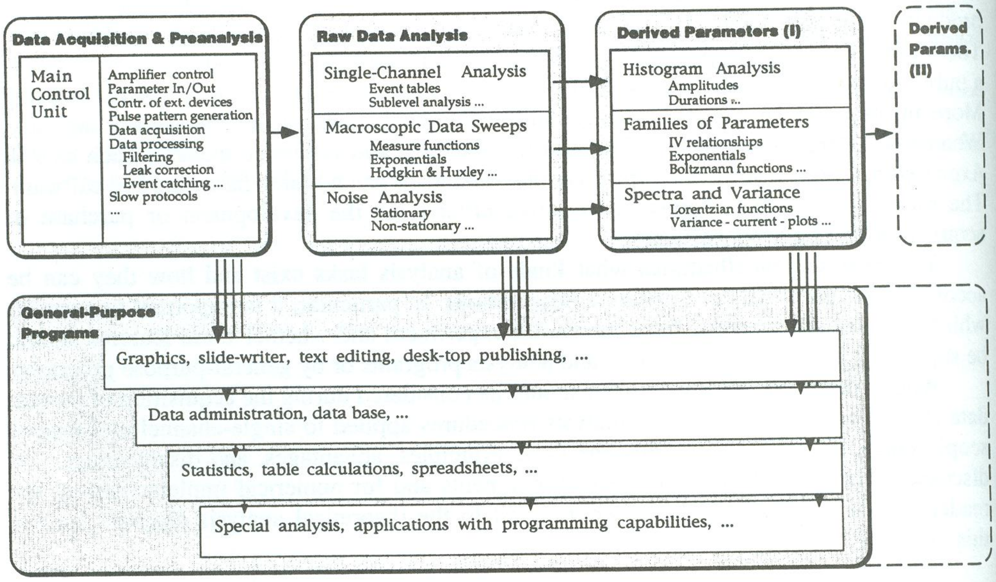

Figure 1. Schematic diagram illustrating the first levels of analysis. The arrows indicate data flow: horizontal arrows show data transfer in a highly specified format, allowing fast access and optimal information transparency as accomplished by sets of dedicated programs; vertical arrows represent the data flow in a less specified format (e.g., ASCII tables) to general-purpose programs, possibly at the expense of convenience, speed, and transparency.

whose solution cannot be implemented in standard software packages and that therefore also necessitate general-purpose programs or programming tools.

Besides those features that are absolutely essential for doing a certain kind of experiment, one should not underestimate the importance of program features that make life easier and which help to avoid errors. In general, the experimenter should have an overview of parameters and control functions that are necessary for decision making during an experiment or during the course of analysis. This includes that the experimenter should get the "right" impression of the data already recorded; i.e., one has to get a feel for the quality of the data. This can be achieved by proper display of the data traces and the structure of the recorded data file as well as by access to on-line analysis options that help one to decide whether to discard records right away or whether to continue with the experiment. Despite the usefulness of having information available at any time, it is very important to be able to "forget" about other parameters that are not currently so important but that may become so at a later stage. This is achieved by a proper configuration of the input parameters, which are stored in the background together with the data so that the experimenter does not need to worry about them during the experiment. Just consider the number of mistakes one can make if the gain setting of a patch-clamp amplifier has to be typed into the computer (or even noted on a piece of paper) after each change. Effort spent in automating such things and designing an appropriate user interface not only saves time but also reduces the error rate considerably and thus has a substantial impact on the success of the work.

Many problems in electrophysiology occur frequently and require specialized analysis features involving considerable programming. In such cases, dedicated analysis software is recommended. This has the advantage that it can be optimally adapted to the structure of the raw data, is commercially available, and is usually faster than general-purpose programs. In Fig. 1 dedicated software is grouped according to levels of complexity, starting with the analysis of raw data (e.g., current versus time). At the next level, parameters derived during primary analysis or transformed data (e.g., event histograms, power spectra) are analyzed further. More levels can be added as necessary. In the following sections some applications of raw data analysis, including analysis of single-channel events, of macroscopic relaxation experiments, and of current fluctuations, are discussed.

Another important consideration is the interface to analysis programs. One should try to use only a few programs that allow easy access to the data generated. This will not be a problem if there are dedicated analysis programs with a similar layout to the acquisition program and that support the same kind of data structure. The acquisition program would be responsible for the experiment control and the recording of pulsed or continuous data together with all parameters necessary to completely reconstruct the experimental configuration, including remarks made by the experimenter. One or several specialized analysis programs (e.g., single-channel analysis, pulse data analysis, noise analysis) would then read such data directly, thereby also providing access to the accompanying parameters. Particularly after specialized analyses, data should be exported as spreadsheet files in ASCII format or other more condensed formats to multipurpose curve-fitting or graphics programs for particular applications.

1.2. Analysis Starts before the Experiment

Signal theory provides a great variety of algorithms and methods for processing electrophysiological data and, in particular, eliminating or to compensating for distortions of the signals. As detailed in later sections, several methods have been developed over the years for the reconstruction of single-channel current events when the recorded signals are obscured by background noise or when too few data samples were recorded. However, these are always time-consuming and approximate methods. Similarly, a lot of effort is usually required—if it is possible at all—to reconstruct the exact experimental conditions at analysis time when one forgot to note the parameters, or if it is difficult to decipher the handwriting in a notebook. Yet another problem is the formating of data so that they can be recognized by several different analysis routines. Therefore, it is a good idea to plan an experiment or a group of experiments very carefully before starting to record data.

Such general experimental design should consider the kind of parameters to be measured and the precision required. The projected task will set limits to the experiment and to the analysis procedures that may not be immediately obvious. Important questions to ask are: Do the data actually contain the information needed for the derivation of the desired parameters? Are there software tools available to extract the desired information, and in which form do these software tools accept the data? The last, but not least important question is how conveniently data acquisition and analysis can be performed.

2. Data Acquisition and Preanalysis

Data acquisition and preanalysis are usually part of complex software packages that consist of main control units for regulating the program flow and thereby the execution of an experiment. In early versions of such programs, pulse pattern generation, stimulation, and the recording of current traces were the major tasks, but many more complex functions, such as versatile displays, on-line analysis, and analysis at later stages, have been added as the applications became more demanding.

2.1. Data Acquisition

Electrophysiological signals are usually transformed to analogue voltages by an amplifier unit. These voltages have to be recorded for display purposes and for later analysis. Analogue signals can be recorded using, for example, an oscilloscope, a chart recorder, or a frequency-modulated magnetic tape. However, because the data should ultimately be imported into a computer for analysis purposes, sooner or later the analogue signals have to be converted to digital numbers. In this section this analogue-to-digital conversion, the filtering of data, and the storage and retrieval of electrophysiological data are discussed.

The sampling of raw data and some signal processing, such as digital filtering and compression, constitute only a small part of acquisition programs. Issues that must be considered in this context are the maximal/minimal sampling speed, the maximum number of sample points that can be output and sampled at any one time, and whether and at what rate continuous data recording is supported.

2.1.1. Analogue-to-Digital Conversion

Because the digitization of data for internal representation on a computer is an important issue, one should consider this step very early in the planning stage of an experiment. Digitization is achieved using an *analogue-to-digital converter* (AD converter), which is connected to the computer bus. In some cases an AD converter is already built into the amplifier (EPC-9, HEKA Elektronik) or into a recording unit (pulse code modulator, PCM; or digital audio tape, DAT; see below).

2.1.1a. ***Dynamic Range.*** Usually AD converters accept analogue voltages in the range of ±10 V, but, because the data will be manipulated using a digital system, AD converters are very often scaled for a maximum range of ±10.24 V. Therefore, the signal output of the amplifier must not exceed these values. On the other hand, the signal should span as much of this voltage range as possible in order to increase voltage resolution. A measure for the efficient use of an electronic device (e.g., amplifier or AD converter) is the dynamic range, which is related to the ratio of the largest and the smallest signals that can be measured. Consider, for example, a patch-clamp amplifier that can register, at the highest gain (1000 mV/pA) and at a given bandwidth, signals as small as 10 fA. Because of the output limit of 10 V, the maximum measurable signal will be 10 pA. This gives a dynamic range of 20 · log(10 pA/10 fA) dB = 60 dB (decibels). To increase the dynamic range, patch-clamp amplifiers have built-in gain functions. With a maximal signal of 200 pA (at the lower gain of 50 mV/pA), and the minimal signal as mentioned above, the amplifier with the reduced gain now has a dynamic range of 86 dB. For the measurement of even larger currents one would need to change the feedback resistor of the amplifier headstage, which in turn would reduce the resolution as a result of an increase in the current noise contributed by the feedback resistor.

Given the dynamic range of the amplifier, we now have to consider the voltage resolution of the AD converter. Usually, the converter represents a voltage level in the range of ±10.24 V as a number composed of 8, 12, or 16 bits; 8-bit boards have now been largely replaced by 12-bit technology, leading to a resolution of 5 mV/bit. Modern converters offer 16 bits, but the two highest-resolution bits are usually obscured by instrumentation noise, such that they actually offer an effective 14-bit resolution (1.25 mV/bit). For some extreme applications, converters with even higher resolution can be used at the expense of sampling speed. An AD board with an effective 14-bit resolution provides a dynamic range of 84 dB [$20 \cdot \log(2^{14} - 1)$], which is well matched to the dynamic range of patch clamp amplifiers.

Nevertheless, care has to be taken that the signal never saturates during data acquisition. One possible problem is the saturation of the amplifier. Although saturating low-frequency components are easily detected, saturation of very fast components is not so readily apparent. Some amplifiers, therefore, provide a clipping monitor and internal low-pass filters that can be set so as to avoid saturation. The aim then is to generate a signal at the output of the amplifier that lies within the range of ±10.24 V and has the highest bandwidth possible without ever saturating the amplifier. This output is then filtered to accommodate the requirements of the AD conversion as specified below.

2.1.1b. ***Aliasing.*** Because the AD converter has only a limited sampling rate, the signal must also have a limited bandwidth. The sampling theorem (Nyquist, 1928) states that the sampling rate should be faster than twice the highest-frequency component within the signal. An ideal sine wave of 1 kHz, therefore, can only be sampled safely if the sampling frequency is greater than 2 kHz. Violation of this principle will cause a distortion of the signal, which is commonly called *aliasing,* as illustrated in Fig. 2. In the frequency domain, this is equivalent to "folding" of higher-frequency components into the frequency range accessible by the sampling device. Put more simply, such a distortion is equivalent to the appearance of low-frequency beating when one combines two high-frequency tones that are not of exactly the same frequency. The problem of aliasing can be quite serious and has to be solved by correct low-pass filtering of the data, i.e., by the elimination of high-frequency components from the signals before AD conversion.

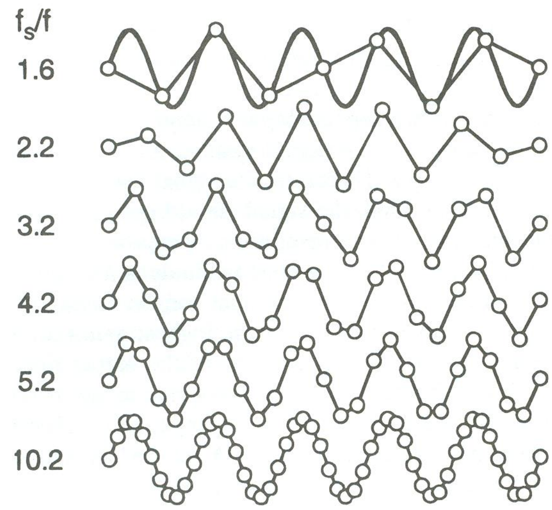

Figure 2. A sine-wave signal of frequency f (thick curve) is sampled at a frequency f_S. The sample points are indicated by circles, which are connected by straight lines. Although the sampling theory defines 2 as a minimum for the sampling factor, f_s/f, it is clearly seen that a reliable reconstruction of the signal is only possible with f_s/f greater than 4 if only a linear interpolation method is used. For small sampling factors not only is the signal reconstructed incorrectly, but inappropriate low-frequency signals also become apparent, an effect referred to as aliasing.

2.1.2. Filtering

Low-pass filtering of electrophysiological signals basically serves two purposes: the elimination of high-frequency components to avoid aliasing and the reduction of background noise in order to increase the signal-to-noise ratio.

The characteristics of low-pass filters are specified by a corner frequency, a steepness, and a type. The corner or cutoff frequency is defined as the frequency at which the power of the signal falls off by a factor of 2; i.e., the amplitude decreases by $1/\sqrt{2}$. This corresponds to an attenuation of the amplitude by -3 dB. This corner frequency may deviate from what is written on the front panel of a filter because some manufacturers use different definitions for the corner frequency. If there is uncertainty about the corner frequency of a certain filter, one should verify it with a sine-wave generator by feeding a sine wave of fixed amplitude into the filter and increasing the frequency until the amplitude of the filter output decays to $1/\sqrt{2}$ of the input signal.

The steepness of a filter is given in dB/octave, i.e., the attenuation of the signal amplitude per twofold increase in frequency (steepness given in dB/decade reflects the attenuation per tenfold increase in frequency). The steepness of a filter function is also characterized by its order, or equivalently by the number of poles. A four-pole low-pass filter has a limiting slope of 24 dB/octave or 80 dB/decade. Commonly used filters have four poles; eight-pole filters are more expensive but also more efficient. In addition, higher-order filters are better approximated by Gaussian software filters, which makes the theoretical treatment during analysis easier (see Chapter 19, this volume). Given the characteristics of a low-pass filter, one can easily see that a certain fraction of the input signal spectrum that exceeds the corner frequency will still be output by the filter. Therefore one has to use a sampling rate greater than the Nyquist minimum (twice the corner frequency). The ratio of the sampling frequency and the corner frequency of a signal is called the oversampling factor. For most applications an oversampling factor of 5 is sufficient if an eight-pole filter is used. If the exact waveform of a signal is to be reconstructed during analysis without interpolation, an oversampling factor of 10 may be more appropriate (see Fig. 2). Figure 3 illustrates how the steepness of a filter at a given corner frequency affects the response to a step input pulse in the time

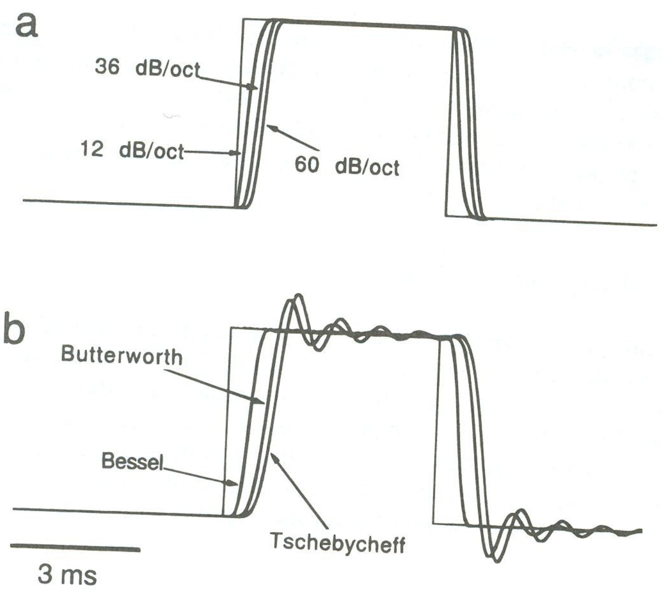

Figure 3. Analogue filtering of a square pulse of 5 msec duration. Data were sampled at a rate of 100 kHz after filtering at 1 kHz with the indicated characteristics. a: Bessel filter with a steepness of 12, 36, and 60 dB/octave. b: Bessel, Butterworth, and Tschebycheff filters with a steepness of 48 dB/octave. Bessel filters introduce the least ringing and are therefore used for single-channel analysis (see Section 3). Butterworth and Tschebycheff filters are inappropriate for single-channel analysis. However, they produce a steeper attenuation of signals outside the pass band and are therefore used for noise analysis (see Section 4).

domain. The steeper the rolloff of the filter function, the less of the high-frequency component will pass the filter; as a result, the onset after the start of the pulse is smoother, and the introduced filter delay is increased.

The most common filter types used in the processing of biological data are Bessel (for time-domain analysis) and Butterworth (for frequency-domain analysis). Bessel filters are most important in the time domain because at a given steepness they cause minimal overshoot of the output signal if the input was a step function (see Fig. 3b). In addition, Bessel filters preserve the shape of signals by delaying all frequency components equally. This is particularly important for the analysis of single-channel current events (see Section 3). For noise analysis in the frequency domain, Butterworth filters and Tschebycheff filters are used. They provide a more efficient attenuation in the frequency domain above the corner frequency but cause considerable ringing in a step response (see Fig. 3b).

So far only low-pass filters have been discussed. Similar considerations hold for high-pass filters that only attenuate low-frequency components, as required for the elimination of slow signals such as drifting baselines. They are used, for example, for noise analysis where only the fluctuating component of the signal is important and slow drifts in the mean current would cause signal saturation. These filters behave as AC coupling devices with an adjustable frequency response. In some cases a combination of both low- and high-pass filters is required. Such band-pass filters are used, for example, to set a frequency window in an amplifier in which the rms noise is measured. A filter that attenuates only a certain frequency band is called a band-reject filter. Very sharp band-reject filters centered around 50 or 60 Hz can be used to eliminate signal distortions caused by line pickup.

As shown above, digital filtering is used mostly for display and analysis purposes. However, if the computer and its peripherals are fast enough, one can always sample at the maximal rate using a fixed antialiasing filter before AD conversion and then use a digital filter for analysis at a specified bandwidth. To reduce the amount of data stored to disk, data compression can be performed after digital filtering if no important information is expected in the high-frequency range.

2.1.3. Control of Experiment Flow

An acquisition program should by today's standards not only record data but also provide a versatile toolbox that enables an experiment to be designed in advance. This is achieved in part by allowing the storage and retrieval of program configurations that meet specific requirements and by macro programming facilities, i.e., "recording and replay" of frequently repeated program events. Important features include parameter input from external devices as well as from the amplifier. In particular, the settings of the amplifier should be recorded as completely as possible so as to reconstruct the experiment at analysis time as precisely as possible.

2.1.4. Pulse Pattern Generation

The most essential part of a program for pulsed data recording is a pulse pattern generator, i.e., the program that sets up the pattern of the voltage-clamp command to be delivered to the amplifier. There should be the possibility of linking various pulses to form a family and linking individual families together to create complicated stimuli necessary, for example, to study the activation and inactivation behavior of voltage-dependent ion channels.

In addition to pulse patterns for the amplifier command voltage, other external devices such as valves or flash lamps can be controlled by pulse generators. This is achieved either by setting digital trigger pulses at specific times during the main pulse pattern or by supporting more than one analogue output channel with separate but synchronized voltage patterns.

2.1.5. Display of Traces and Relevant Information

Because of the inherent instability and variability of biological preparations, an electrophysiological experiment is a highly interactive process; decisions often have to be made during the course of a measurement. It was mentioned above how important it is to have facilities that allow the configuration of stereotypic tasks. This is no contradiction to the required flexibility, since such simplifications give the experimenter time to concentrate on the measurement, to judge incoming data, and to decide what to do next. Therefore, the presentation of data during an experiment is an important issue. Fast graphics allow the display of current traces on the computer screen almost in real time, mimicking an oscilloscope. Additional features make such display procedures superior to an oscilloscope, for example, display of traces with/without leak correction and with/without zero-line subtraction, display of leak currents and other traces that were recorded in parallel, the ability to store previously recorded traces, and comparison of incoming traces with a specified reference trace. Despite these advantages, however, a high-speed oscilloscope cannot be entirely replaced by a computer, for the traces as displayed on the screen give only a representation of the real data, depending on the coarseness of the AD conversion and the resolution of the computer display. As a result, one may not always see very fast spikes because of capacitive transients and may therefore introduce errors through signal saturation.

Besides having the current traces being displayed as realistically as possible, it is important to have access to the values of other experimental parameters, which are stored together with the raw data. Because the requirements for external parameters may vary considerably depending on the kind of experiment, it is best if only those parameters actually of interest are brought to the attention of the experimenter. The importance of data documentation cannot be underestimated. Acquisition and analysis programs should therefore provide text input for individual sweeps, families, groups of families, and the entire data file. Marks at individual data sweeps, e.g., indicating blank sweeps in nonstationary single-channel recordings, are very helpful and save a lot of time during later stages of analysis.

Increasing computer performance and decreasing data storage costs encourage the tendency to store more data than actually required. Worst cases are the storage of obviously redundant data and "bad" data that do not contain any valuable information. Although one can always say that such data can be deleted at a later stage, it should be stressed that it becomes increasingly difficult to extract a fixed quantity of information from an increasingly large data set. In other words, it takes more time to find a needle in a haystack the larger the haystack is. For this reason, acquisition and analysis programs should provide access to data recorded during the experiment to facilitate selective data deletion, compression, and averaging.

2.1.6. On-Line Analysis

The above-mentioned display features already perform a kind of data analysis. More quantitative analysis can be supported by dedicated analysis routines that process incoming data traces according to specified procedures. Examples of such on-line functions are the determination of values such as minima and maxima, time to peak, mean, variance, etc. These parameters will be determined within a specified time window conveniently set with respect to a relevant segment of the data trace, as specified in the pulse generator. The analysis results can then be displayed as numerical lists or as plots. In the latter case, several abscissa options allow the versatile generation of on-line analysis plots. For example, in a steady-state inactivation curve, one would determine the peak current within the constant test-pulse segment; the results would then be plotted as a function of the voltage of the conditioning prepulse segment, which was previously specified as a relevant abscissa segment in the pulse generator.

2.2. Data Storage and Retrieval

Long traces of single-channel recordings take up a lot of disk space and are therefore often stored in an inexpensive medium before analysis is performed section by section using a computer. Analogue signals can be stored on analogue magnetic tapes in a frequency-modulated format. Nowadays, however, signals are more often digitized before storage by a PCM and then stored on video tape using a conventional video cassette recorder (VCR)

(Bezanilla, 1985) or on digital audio tapes using modified DAT tape drives (Fig. 4a,b). These media are inexpensive and capable of storing sampled data at rates of up to 100 kHz for more than an hour. Alternatively, one can read data into the computer and transfer them directly to an inexpensive storage medium using a DAT drive hooked up to the computer. Optical WORM (write once, ready many times) disks are still rather slow but offer, like other disks, random-access capability. In most cases storage on exchangeable disks, which can be accessed like a conventional hard disk, is more practical because of shorter access delays. Hard disk cartridges or magnetooptical disks have come down in price considerably, such that they are now a feasible alternative for mass storage of electrophysiological data. For high-speed requirements these media may still be too slow, and a large hard disk may be required as an intermediate buffer.

The choice of long-term storage medium is mainly determined by the amount of data that has to be stored. If one wants to store single-channel data filtered at 10 kHz at a time resolution of 44 kHz (typical in PCM/VCR systems), one could store data for the entire running time of a video tape (e.g., 90 min). If the data are stored as 16-bit numbers (2 bytes), the sampling process amounts to a data flow of 2 bytes · 44,000/sec = 88 kbytes/sec. This

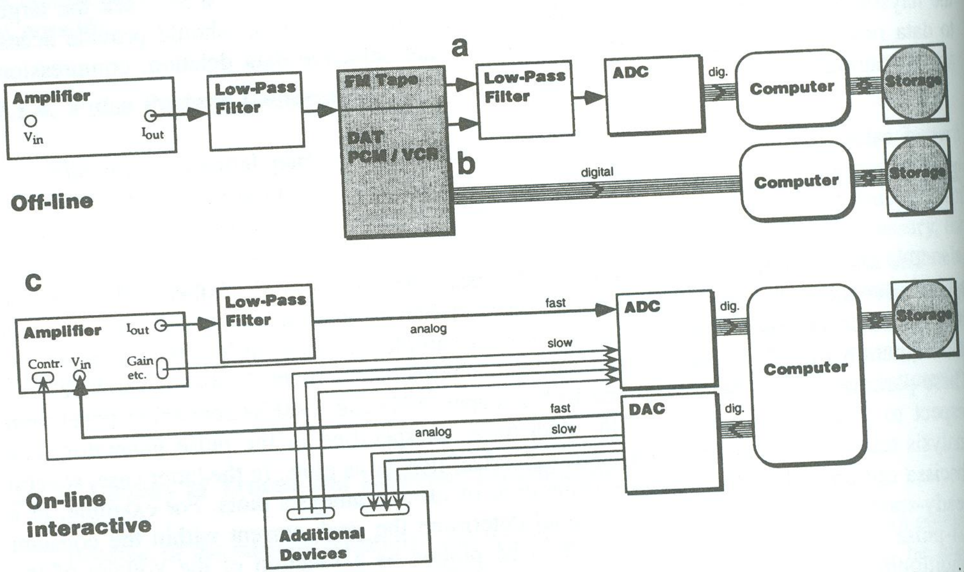

Figure 4. Possible configurations for the acquisition of patch-clamp data. In a and b the current monitor signal from the patch-clamp amplifier is passed through an antialiasing filter appropriate to the sampling frequency of a tape-recording device (FM tape, digital audio tape, or video tape). a: For off-line analysis data are replayed from the tape, passed through a filter according to the sampling rate of the ADC board, and read into a computer. b: Alternatively, using DAT drives or PCM/VCR combinations, the digitized signals can be transferred to the computer directly in order to reduce errors from a second digitization as in a. c: Configuration for an interactive, semiautomated acquisition system. The computer controls the patch-clamp amplifier and other external devices via a DA interface. The analogue signals may be grouped into fast signals (e.g., command voltage) and slow signals (e.g., setting of amplifier gain, perfusion valves). Similarly, analogue signals are acquired on a fast (e.g., current traces) and a slow (e.g., amplifier gain, filter setting, temperature) time scale. Further automation is achieved with the EPC-9 patch-clamp amplifier (HEKA Elektronik, Lambrecht, Germany) in which all functions are directly controlled by digital connection to a computer. In c, long-term data storage capabilities are mandatory, whereas in a and b the tapes can also be used as long-term storage media. Only selected sections to be studied more thoroughly at a later stage would be stored to disk or to another computer memory.

means that a recording of 13.6 sec could be stored on a floppy disk (1.2 Mbyte). An exchangeable hard disk (80 Mbytes) would be full after 15 min, a magnetooptical disk (250 Mbyte) after 47 min, and a DAT streamer tape (2 Gbyte) after 6 hr 18 min.

Low-activity single-channel recordings can be compressed considerably if only events of interest are stored. For this purpose data are sampled, passed through an event catcher, and stored selectively. The same kind of software can be adapted to the recording of spontaneous synaptic potentials or currents, for example.

Short sections of data, and especially those recorded under varying conditions, such as during a voltage-step experiment, should be stored directly on the computer's hard disk (see Fig. 4c). This is the easiest way of solving the problem of exact timing. Several software packages are available, depending on the hardware, that provide the user with many tools for keeping track of the experiment and for data analysis.

2.3. Interface to Other Programs

The kernel of an acquisition program not only controls the flow of an experiment but also defines the data structure. Because the tasks and approaches considered during the development of acquisition software vary, most programs generate their own specific data structure. Analysis programs should therefore recognize a variety of structures for data import; however, this is rarely the case. Programs that convert one data structure into another are available from several companies, but the same problems arise if the information contained in the two data structures is not compatible. It is usually possible to convert important information such as raw data traces, timing, and gain, but the conversion of accompanying information and the exact reconstruction of the pulse pattern used to evoke the stored data are often complicated or simply impossible. Therefore, it is advantageous to use dedicated analysis software (see below) from the same source as the acquisition program. Sometimes, however, this will not be possible because some specific analysis features may not be supported or may prove to be insufficient. In such cases, and for interfacing to general-purpose programs (see below), acquisition programs must allow data output in a more general format, for example, as text tables (ASCII).

3. Single-Channel Analysis

Current recordings from membrane patches with only one or a few active ion channels are suitable for the analysis of opening and closing current events. A direct transformation of the current signals, as a function of time, into amplitude histograms on a sample-to-sample basis can be used to obtain an overview of the single-channel records. Evaluation of the peaks in the histograms enables the number of single-channel current levels or the existence of sublevels to be determined.

The major task of single-channel analysis programs is to compile event lists. For that, single-channel current events have to be detected, quantified, and stored. Before the actual event analysis can be performed, the data have to be filtered so that they are suitable for this type of analysis. The output of an event-detection program is an event table that contains for each transition at least a level index, an amplitude, and a duration.

At the next level of analysis, information has to be selectively extracted from these event tables. Typical applications are the compilation of amplitude and duration histograms for certain event transitions or levels, respectively. Such histograms can be compiled and displayed in various ways and are used to fit model functions, such as Gaussian curves for amplitude histograms or multiexponential functions for lifetime histograms, to the data.

Single-channel analysis packages are commercially available for various types of computer hardware. Several other programs of this kind have been developed in various laboratories but are not commercially available. Great differences in the quality and capabilities of these programs make necessary a very careful evaluation of the program specifications or of a demo version before purchase. The following topics may help to define the requirements of the features provided by the programs.

3.1. Data Preparation

3.1.1. Digital Filtering and Data Display

Programs should take in continuous as well as pulsed single-channel data. For display and event detection, data have to be presented on the computer screen after passing through a digital low-pass filter. Usually Gaussian filters are used for this purpose (see Fig. 5) because they resemble a Bessel characteristic of high order, which is mandatory for single-channel analysis (see Fig. 3). In addition, they can be described in a mathematically compact form (see Chapter 19, this volume) and therefore can be implemented in software that executes relatively fast. Nevertheless, digital filtering demands a lot of processor time, and fast computers are therefore preferred.

Individual channel transitions have to be inspected at high time resolution, and this can result in a loss of overview when one is analyzing a long data record. Therefore, two or three simultaneous data representations at different time scales are very helpful.

At the highest time resolution, data points may appear very sparse on the screen. For a smoother appearance and for a careful analysis of single-channel records, the individual data points have to be connected using some kind of interpolation. The easiest method is to connect the values by straight lines. A better approach is to use more complicated spline functions such as cubic polynomials (see Chapter 19, this volume).

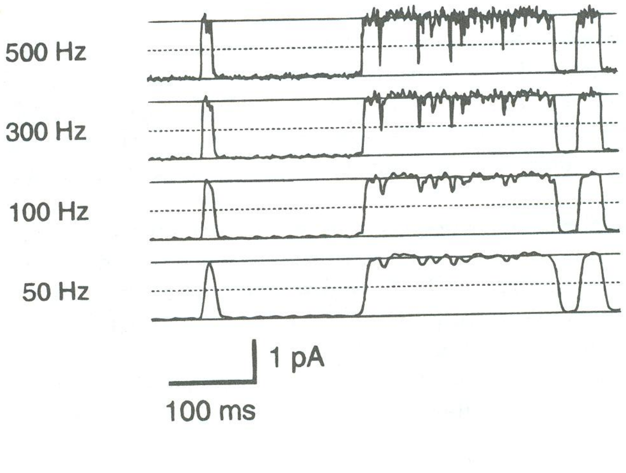

Figure 5. Single-channel current events were recorded at a rate of 2 kHz and then passed through a digital Gaussian filter with the indicated corner frequencies. With an appropriate filter setting, the brief closures during the long channel openings can be reduced in amplitude such that they do not cross the 50% threshold (dotted line).

3.1.2. Leak Correction

If single-channel events are elicited by voltage pulses, the related capacitive current transients and leak currents have to be compensated before actual single-channel analysis can be done. During the recording of nonstationary single-channel records with low activity, data sweeps without channel openings can be averaged and used as a background reference. The average of these "null" traces can then be directly subtracted from the individual data sweeps. If the noise of the averaged leak records is too large, it can be idealized by fitting theoretical functions to it. Usually polynomials or summed exponential functions are quite appropriate.

3.2. Event Detection

For single-channel analysis, an event is defined as a sudden change in current as the result of the opening or closing of an ion channel. Once detected, such an event has to be characterized and stored in an event table, which is then used for statistical analysis of channel currents (amplitudes) and channel kinetics (durations). An event has to be characterized by at least two parameters: (1) an amplitude and (2) a time (e.g., time when the amplitude reaches the 50% level). In order to facilitate analysis, one also notes a current level index, which specifies how many channels are open after (or before) the event happened (e.g., 0 = all channels closed, 1 = one channel open). Besides these "regular" events one can introduce "special" events such as sublevels relative to a normal channel opening. This would mean that a program for event table analysis could extract all sublevel durations as well as all main level durations with and without sublevel contributions. Additional information might include the current variance and an indication of how the event was detected and how the amplitude was determined (e.g., amplitude determined manually or automatically, or amplitude taken from the previous event). In order to avoid the effects of drifting baselines, two amplitudes can be stored for each event in addition to the durations used to determine them. Particularly for nonstationary single-channel event tables, the event timing must be given with respect to the start of the corresponding stimulus or to a specified pulse segment.

The main part of single-channel analysis programs consists of event-detection algorithms. Several strategies are followed in commercially available programs. Most of them rely on automatically or manually predefined baseline currents and single-channel current amplitudes; transitions are then determined as the crossing of certain critical current thresholds. Automated methods often make use of variance measurements in sliding windows in order to obtain objective criteria for threshold crossings.

3.2.1. Filtering for Single-Channel Analysis

For stationary and nonstationary recordings the hardest task is choosing the right bandwidth for analysis. This is largely determined by the signal-to-noise ratio of the records; i.e., to what extent does background noise result in the detection of false events? On the other hand, too narrow a bandwidth may cause short channel events to be missed. Because the latter aspect can only be judged if one knows what the signals look like, it may be necessary to perform a preliminary single-channel analysis at an estimated bandwidth. The distribution of the measured events will then provide an estimate of how many events were missed, and an optimal bandwidth can then be chosen (see Chapter 19, this volume).

3.2.2. Threshold-Crossing Methods

The most popular method for event detection uses a threshold that is halfway between the open and the closed current levels (see Fig. 5). These "50%-threshold" methods are easily implemented and do not require correction of event durations as long as the current reaches the full channel level, because the effect of filter delay on the time lag between the actual channel transition and the measured time at which the current signal crosses the threshold is the same for openings and closings. For events that do not fully reach the next level, a correction has to be applied (see Chapter 19, this volume). One can test the correction methods applied in single-channel analysis programs by creating artificial square-shaped single-channel events and filtering them with a filter rise time of approximately 75% of the event width. The events are then analyzed with a 50%-threshold criterion by setting the correct single-channel amplitude. An underestimation of the actual event width is an indication that the correction has not been properly implemented.

In nonstationary recordings the timing of the events has to be correct with respect to the start time of the sweep in order to yield correct first-latency intervals. All detected events therefore have to be shifted to the left by the delay introduced by the system response, including that of the digital filter used for analysis.

Threshold-crossing methods with higher than 50% levels can be used for very noisy data, although stronger filtering would be better. Levels of less than 50% can be necessary for the detection of events in very low-noise data acquired at the maximum attainable rate. Such a limitation may arise if a sampling device cannot take in data at a rate high enough for the actual time resolution of the current recording. Suppose, for example, that one wants to record single-channel events with an amplitude of 20 pA on a PCM/VCR combination with the maximum rate of 44 kHz, and the maximum corner frequency of the steep low-pass filter to be used is 10 kHz. At this bandwidth the rms background noise could be 500 fA. Thus, a detection threshold could be safely set to eight times the standard deviation, i.e., 4 pA, corresponding to a 20% threshold criterion.

3.2.3. Time-Course Fit

Threshold-crossing methods yield good results if the single-channel events are of homogeneous amplitude. When, for example, brief flickers during a burst of activity represent full channel closures, threshold-crossing events can be safely converted into event durations. Ambiguities arise if one is not certain whether the events really are of "full" amplitude, i.e., if they are too short for satisfactory amplitude determination. In such cases an idealized square-shaped event for which the amplitude and duration are allowed to vary is fitted to the data in order to enhance the resolution and reliability of the analysis (Colquhoun and Sakmann, 1985; Colquhoun, 1987; Chapter 19, this volume). For the creation of an idealized channel event, the effect of filtering according to the system transfer function must be taken into account. The time course of a step response can therefore be approximated by Gaussian-filtered square events; alternatively, measured step responses can be used. Note that this method is usually more time-consuming than simple threshold-crossing methods.

3.2.4. Automatic Data Idealization

Single-channel event detection can be automated in several ways. A kind of semiautomated method is to measure several single-channel events manually, to get an idea of the single-channel amplitude, and then to use this estimate to define a threshold-crossing criterion for an automatic search routine. Starting from a baseline segment, this would identify the first transition, measure the amplitude after the transition if the open duration is long enough, reset the amplitude of the open channel to the measured value, and continue to search for the next event (closing of the present channel or opening of another one). Sublevel events and fast flickers can cause serious problems with such algorithms. Therefore, one may use automatic methods of this kind only if (1) the data are ideally suited or (2) the experimenter observes the automatic process and interrupts it if the algorithms start to catch false events. In either case, it is very helpful if already stored data idealizations (i.e., contents of the event lists) can be superimposed on the measured data at any time. In this way the experimenter can reconfirm that the automatic algorithms yield sound results.

Several other algorithms have been developed that use the mean current and the variance in sliding windows for the detection of channel transitions. Such edge detectors can be iteratively applied to the raw data so as to optimize event detection (e.g., Sachs *et al.,* 1982; Kirlin and Moghaddamjoo, 1986; Moghaddamjoo *et al.,* 1988; Pastushenko and Schindler, 1993).

3.2.5. Maximum-Likelihood Methods

Very rigorous methods can be applied that directly maximize the likelihood of a kinetic model to describe a certain data set under consideration. Such methods use hidden-Markov algorithms and are computationally very expensive (Chung *et al.,* 1990, 1991; Fredkin and Rice, 1992; Auerbach, 1993). Since neither event detection nor compilation of dwell-time histograms is required, they promise great savings in analysis time spent by the experimenter.

3.2.6. Drifting Baseline Problems

Most of the methods mentioned above work well within the limits of their time resolution if the baseline current does not change. In many applications, however, a change in seal resistance during a long recording period cannot be avoided. The user therefore has to ensure that during single-channel analysis the actual channel-closed period is taken as the baseline. This is straightforward if the channel activity is not too high and if there are enough long-lived closed periods. Then the current levels of periods between channel openings can be used frequently to determine the new baseline (e.g., as the mean or the median of a specified number of data points). Baselines between two baseline determinations then have to be interpolated. There are also more involved methods for automatic baseline tracking (e.g., Sachs *et al.,* 1982), but the user should always verify that the algorithms did not mistake a long-lived open channel state or a substate for a new baseline.

3.3. Analysis of Histograms

After the compilation of event tables, specific information such as single-channel amplitudes and open and shut periods have to be extracted for further analysis. For display purposes, and also for comparison with theoretical predictions, the collected event information is displayed in the form of histograms. The abscissa of a histogram is divided into intervals (bins) of event amplitude or event duration for amplitude or lifetime histograms, respectively.

The number of events within a certain observation period that fall within the individual bins are counted and displayed as bars.

3.3.1. Display of Histograms and Binning Errors

There are several methods for displaying dwell-time histograms. A straightforward way is to use linear scaling for durations and the number of entries per bin. Since dwell-time distributions are usually sums of exponential functions, kinetic components are more easily appreciated if an exponential time base is used (McManus *et al.,* 1987). A very useful presentation of dwell-time histograms displays the square root of events in bins of exponentially increasing width (Sigworth and Sine, 1987; Jackson, 1992). In the case of a single-exponential distribution the *probability density function* (pdf) then peaks at the time constant, and all bins have the same theoretical scatter throughout the entire time range.

Several errors can be introduced when a histogram is constructed. Some of them are related to the discreteness of the sampled signals. If the bin boundaries are not multiples of the smallest resolvable unit (current or time, respectively), some bins may have a higher chance of being populated than others. In amplitude histograms this problem is largely eliminated if single-channel currents, based on the average of many sample points, are stored as real numbers. For dwell-time histograms this error can be serious if the exact timing of the channel-open or -shut times is based on individual samples. The problem becomes minimal if an interpolation method (e.g., a cubic spline) for the exact determination of the threshold crossing is used.

Noise in a signal that is to be displayed as histograms also causes a distortion of the results, particularly if the event distribution is highly nonlinear. Such an effect can be accounted for but is usually of minor importance (Chapter 19, this volume).

Another kind of binning error arises for large bin widths when one uses the center of the bins to compare heights directly with theoretical probability density functions (McManus *et al.,* 1987). This problem can be avoided by following a more rigorous approach using probability distribution functions (e.g., Sigworth and Sine, 1987).

3.3.2. Fit of Theoretical Functions to Histograms

There are various methods for fitting theoretical functions to histograms. They range from very simple least-squares fits to the histogram bins to the use of maximum-likelihood methods (Chapter 19, this volume; Magleby, 1992) on bins as well as on the events themselves. In the latter case, binning errors are completely eliminated. For the optimization itself various algorithms, such as simplex, steepest descent, or Levenberg–Marquardt methods, are used. For discussion of these methods and the estimation of error bounds, see e.g., Dempster (1993). An important issue for the fitting of binned distributions is the weighting of the data points. Considerable improvements over fits of linear dwell-time histograms are yielded by fitting histograms with logarithmically scaled bin widths (e.g., McManus *et al.,* 1987; Sigworth and Sine, 1987).

3.3.3. Compilation of Amplitude Histograms

The compilation of event amplitude histograms by specification of event level range and bin width is implemented in most single-channel analysis packages. An important requirement for analysis programs, however, is that they allow the selection of single-channel transitions under certain conditions in order to enhance the accuracy of analysis and to restrict analysis to subsets of the data. Here are several criteria that might be considered for the extraction of entries from event tables:

- *Event level.* Events from the baseline to the first open level can be measured most precisely because the noise level increases as more channels are open. Thus, for the compilation of an amplitude histogram, one may want to discard transitions between higher levels than the first one if there are enough events of this kind available.
- *Event duration.* Only if the duration of the closed and open time before/after the transition is long enough can a precise amplitude measurement be achieved. The precision of an amplitude histogram is therefore increased if channel events shorter than a certain duration are discarded.
- *Detection method.* In some programs both automatic event detection and amplitude measurement are supported. One might select for an amplitude histogram only those events that were manually measured or at least visually validated by the experimenter. In such cases one does not rely on events that were measured by an automatic algorithm in a possibly inappropriate way.
- *Time range.* In stationary single-channel analysis, one might want to compare events from early and late parts of the recordings in order to check for drift phenomena or effects such as the shifting of gating mode (e.g., Zhou *et al.,* 1991). In nonstationary recordings, separation of events from early and late phases of individual pulses might help to separate kinetically distinct channel components.
- *Sublevels.* If sublevels were marked as "special events" (see Section 3.2), there must be a separate way of selecting them. In order to address the question of when a sublevel occurs preferentially, it is helpful if one can extract them conditionally by specification of the previous/next event before/after the sublevel.

3.3.4. Gaussian Distributions

Usually one or more Gaussian distributions, each characterized by a mean value, I_0, a variance, σ^2, and an amplitude, a, are fitted to the histograms manually or by least-squares or maximum-likelihood methods.

$$n(I) = \sum_{i=1}^{n} \frac{a_i}{\sqrt{2\pi}\,\sigma_i} \exp\left(\frac{-(I_i - I_{0i})^2}{2\sigma_i^2}\right) \qquad (1)$$

Deviations from Gaussian functions are discussed in Chapter 19 (this volume). Similarly to dwell time histograms, maximum-likelihood methods can be employed to fit functions to individual events rather than to the histogram bins.

3.3.5. Compilation and Display of Dwell-Time Histograms

As for amplitude distributions, analysis programs should provide versatile tools for the extraction of dwell times from event tables. Here are several criteria that might be considered for the selection of events:

- *Event level.* Open and closed times can be extracted from event tables. This is trivial if only one channel is active during the recording period. If multiple openings occur, however, errors are introduced for open times if they are skipped or if the superposition of two channel openings is counted as one.

- *Event amplitude.* If two clearly distinct single-channel amplitudes were measured, then the kinetics of one of the components can be characterized separately if amplitude ranges are set for the extraction of events from the event table.
- *Event duration.* For duration histograms the influence of false events resulting from background noise can be reduced if a minimum event duration is set.
- *Burst events.* If an ion channel has two closed states with clearly different dwell times, openings can appear as bursts. Approximate histograms of burst lengths and gaps between bursts can then be compiled by setting a minimum dwell time for a channel closure to be accepted. Given such a minimum gap time, event tables have to be recalculated because the remaining closed and open durations will change. After the definition of bursts, gaps and open times within bursts can then be extracted from event tables. For objective criteria of how to set such minimum gap times see Chapter 19 (this volume).
- *Time range.* During long recordings single-channel activity may change (e.g., Hess *et al.,* 1984). For an overview of stationarity, programs should provide features to display open probability, as determined in a sliding window, as function of time. Similarly, event histograms can be compiled from selected time periods to determine small changes in channel kinetics during the course of an experiment.
- *First latencies.* Nonstationary data require the analysis of first latencies, i.e., the time from a given stimulus until the first channel opening. This is an important parameter for the investigation of inactivating channels when one wants to obtain information on the activation process that is not obscured by the inactivation mechanism (e.g., Sigworth and Zhou, 1992).

More involved statistical analyses, such as the correlation of adjacent intervals (e.g., Blatz and Magleby, 1989), are usually not implemented in standard single-channel analysis programs. In such cases the event tables usually have to be reanalyzed with user-designed software.

3.3.6. Probability Density Functions

The simplest way to determine time constants from dwell-time histograms is to fit a sum of exponential functions, each characterized by a time constant, τ, and a relative amplitude, a (equation 2).

$$pdf(t) = \sum_{i=1}^{n} \frac{a_i}{\tau_i} \exp\left(\frac{-t}{\tau_i}\right), \qquad \sum_{i=1}^{n} a_i = 1 \tag{2}$$

Independently of the binning, maximum-likelihood methods can be employed to fit *pdfs* directly to the events rather than to the histograms (see Chapter 19, this volume).

3.3.7. Missed Event Correction

Because of limited time resolution there are always events missed during the detection procedure. In closed-time histograms the limited bandwidth can, as a first approximation, be compensated for by neglecting the first bins in which the nonresolved events are missing, provided that there are no very fast opening events. If the time constants under consideration are clearly longer than the sampling period, this method yields acceptable results.

Open times, however, are always badly affected by the limited time resolution, because a missed closing event will result in an overestimated open time. Several theoretical methods have been developed to overcome this problem (see Chapter 19, this volume; Roux and Sauve, 1985; Blatz and Magleby, 1986; Crouzy and Sigworth, 1990; Hawkes *et al.,* 1991).

Once a kinetic scheme has been decided on, a rigorous method can be applied, such as that of Magleby and Weiss (1990). This method uses simulated single-channel data that have been masked with background noise, filtered, and analyzed exactly as the experimental data. The resulting simulated histograms, based on simulated data, are then compared with the measured histograms. The same time-consuming process is repeated with altered kinetic parameters of the model until a satisfactory match between measured and modeled histograms is achieved.

3.4. Open-Channel Analysis

So far we have just been concerned with single-channel events that have been identified with a detection method and then characterized by an amplitude and a duration. More information can be extracted from single-channel records when single open-channel currents are recorded at varying potentials or by analyzing the current noise in individual open channels.

3.4.1. Conditional Averaging

If the ion channel open times last several milliseconds, single-channel current–voltage relationships can be acquired by the application of voltage ramps. However, channels may not stay open for the duration of an entire ramp. Therefore, single-channel programs should provide editing features that allow the extraction of data sections from traces for averaging. Such conditional averaging is illustrated in Fig. 6, which shows four single-channel responses to identical voltage ramps. Open-channel sections can be selected with a cursor or mouse-operated routine and are stored in an accumulation buffer. The number of entries per sample point is also stored and then used for the proper scaling of the averaged single-channel current–voltage relationship. Note that after this procedure the errors in the individual data points are no longer the same because of the heterogeneous averaging. Besides the open-channel sections, baseline entries can be stored and used for leak correction.

3.4.2. Open-Channel Histograms

Single-channel recordings can also be analyzed on a sample-to-sample basis by the compilation of current histograms as illustrated in Fig. 7. The peaks in such histograms indicate the main current levels, and the widths of the peaks are a measure of the current noise in the corresponding level. Deviations from the symmetry of the peaks is indicative of nonresolved events that cause a skew in the distribution (see Fig. 7b). The relative lifetime of a current level is obtained from the relative area.

Note that these methods, which use raw data traces, generally require a very stable baseline and the proper selection of data sections. Even small shifts in the baseline current

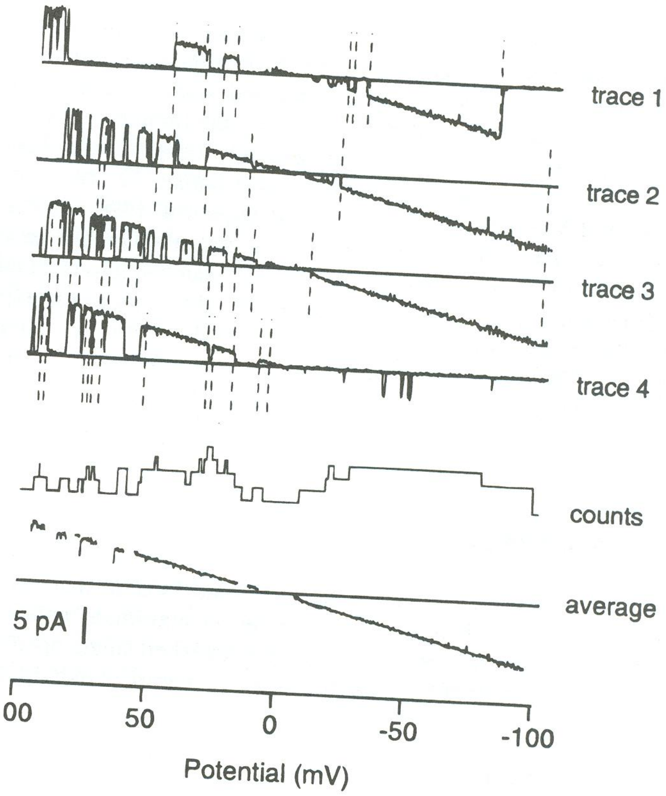

Figure 6. Conditional averaging of single-channel currents evoked by a voltage ramp from +100 to −100 mV of 200 msec duration. Open-channel sections without flickers were selected (vertical dashed lines) and accumulated. The accumulated data were then divided by the number of entries per sample point and displayed as "average," yielding a single-channel current–voltage relationship with some gaps where no open channels could be recorded.

can cause considerable broadening of the open-channel current histograms, which results in an overestimation of the noise.

More information about channel behavior can sometimes be extracted by selecting and analyzing current traces that represent only one channel state (e.g., open or closed). The fluctuating current that is measured during an open-channel period is composed of both statistical background noise and the noise arising from the flow of ions through the channel. Both of them can be described as a first approximation by Gaussian functions (equation 1). Additional processes that generate current noise will add to this Gaussian noise and can cause deviations of the histograms from the typical bell shape.

Long-lasting sublevel events, for example, can be detected directly as separate peaks in the open-channel histograms between the peaks of the main level and the baseline.

If closing events are too brief to be resolved directly they will not show up as separate peaks in open-channel histograms but will skew the histograms in the closing direction (see Fig. 7b). Assuming flicker events to cause complete channel closures, several theoretical approaches can be applied to extract information about the flicker kinetics from the skewed histograms. For heavily filtered signals, Yellen devised a method that uses the fit of β-functions to histograms to yield time constants for flicker dwell time distributions (Yellen,

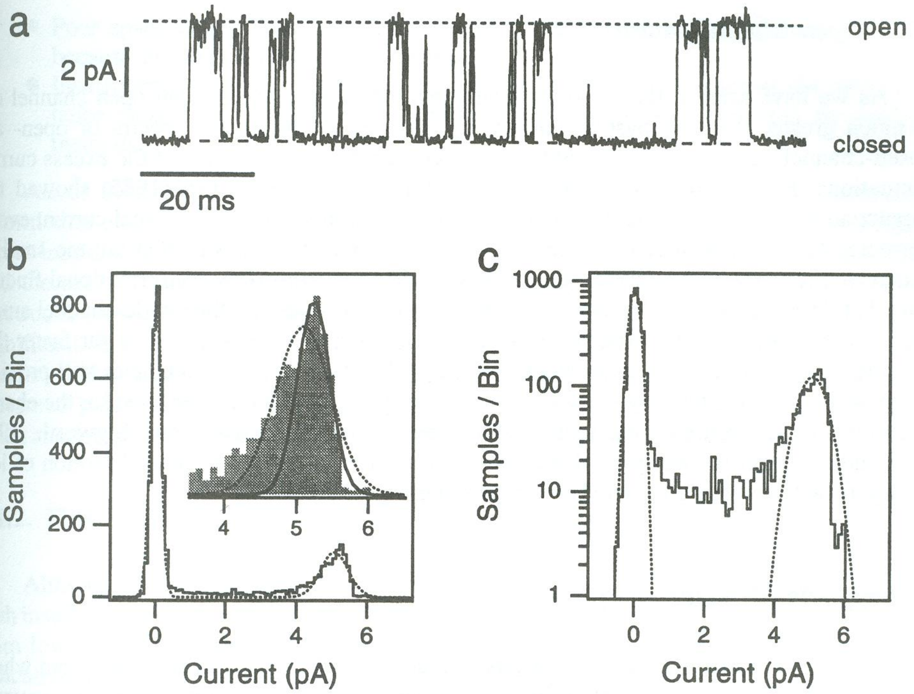

Figure 7. Analysis of single-channel recordings on a sample-to-sample basis. a: Outward currents low-pass filtered at 1 kHz and recorded at a rate of 4 kHz. b: Linear current histogram with a bin width of 0.1 pA. the dotted line represents a fit of the sum of two Gaussian functions (equation 1) to the baseline peak (center, I_{0c}: 0.01 pA, height, a_c: 858 samples, standard deviation, σ_c: 0.147 pA) and the open-channel peak (I_{0o}: 5.09 pA, a_o: 125 samples, σ_o: 0.389 pA). Because the recording is based on the activity of only one channel, the ratio of the areas under the two curves yields an open probability of $[1 + (a_c\,\sigma_c/a_o\,\sigma_o)]^{-1} = 0.28$. Although the baseline peak is reasonably well described by a Gaussian function, the open-channel peak is clearly skewed toward zero because of brief closing events. A better estimate of the peak current is obtained by fitting a Gaussian function only to the data above 5 pA, as illustrated by the solid curve in the inset (I_{0c}: 5.23 pA, a_o: 145 samples, σ_o: 0.226 pA). The current value of the peak of this histogram is indicated as a dashed line in part a. c: The same histogram as in b, but with logarithmic scaling of the ordinate.

1984). The kinetics of very fast and rare closing events far beyond the actual time resolution of the recording system can be estimated from the higher moments of the open-channel histograms (Heinemann and Sigworth, 1991).

3.4.3. Mean-Variance Methods

Particularly useful for the identification of sublevel events are the methods proposed by Patlak (1988, 1993) in which mean current and variance are calculated in a sliding window. Three-dimensional histograms of the number of entries as a function of variance and mean current allow the separation of distinct current levels as peaks whose volume and shape contain information about the kinetics of the events.

3.4.4. Open-Channel Noise

As we have seen in the previous examples, the current noise in an open channel can be much greater than the baseline noise. By analysis of the power spectra of open- and closed-channel currents, more insight can be gained into the properties of the excess current fluctuations. For nicotinic acetylcholine receptor channels, Sigworth (1985) showed that Lorentzian components in the power spectra (see equation 7) of open-channel current events (corrected for baseline spectra) were indicative of current fluctuations that are too small to be detected directly in the time domain. They could arise from slow conformational fluctuations that do not close the channel completely but rather modulate the single-channel amplitude. If the kinetic events responsible for the generation of excess noise are far faster than the time resolution of the measurement, the spectral density at low frequencies still provides information about the underlying processes. These approaches have been used for the characterization of fast, nonresolved, channel-blocking events (Heinemann and Sigworth, 1988, 1989) and even for the analysis of the shot noise generated by the statistical motion of ions as they flow through a channel (Heinemann and Sigworth, 1990).

4. Analysis of Macroscopic Currents

In this section we discuss the analysis of macroscopic currents as obtained from whole-cell recordings or from patch recordings using large pipettes. In most cases ionic currents do not occur spontaneously but must be evoked by stimuli such as a change in the membrane potential or the fast application of agonist. Because voltage-clamp experiments are most common, they are discussed in more detail. The problems of pulse pattern generation, parameter control, leak correction, and, finally, various methods of extracting information about single-channel properties from macroscopic data are addressed.

4.1. Parameter Control in Relaxation Experiments

A relaxation experiment is a type of measurement in which an experimental parameter is changed suddenly in order to perturb the equilibrium of the system under consideration. After the perturbation a new equilibrium will be reached with a time course dependent on the new experimental parameters (e.g., new potential after a voltage step). In order to characterize the kinetics of voltage-dependent ion channels, for example, the membrane potential is changed according to a pulse pattern comprised of a number of segments of variable duration and potential.

4.1.1. Voltage-Clamp Performance

For later analysis of the recorded currents, it is very important to consider how much the potential at the membrane deviates from the desired potential as specified in the pulse protocol. Typical reasons for such deviations are:

- Limitations in the voltage-clamp amplifier (e.g., from filtering of the stimulus at the input of the patch clamp amplifier)

- Poor space clamp (i.e., not all membrane areas can be held at the same potential because of unfavorable membrane topology)
- Insufficiently compensated series resistance (i.e., voltage drop across the series resistance caused by large currents could not be completely corrected by analogue methods.)

All of these problems can be rather serious when one wants to derive quantitative information on channel kinetics from the current recordings. In patch-clamp recordings the errors introduced can usually be easily estimated by considering the filtering of the stimulus and the series resistance compensation (if necessary at all). For recordings with fine-tip electrodes (two-electrode voltage clamp), deviations from the theoretical potential can be significant, but they can be measured by recording of the actual membrane potential in parallel to the membrane current. The estimated real potential profiles (patch clamp) or the measured potential profiles (two-electrode voltage clamp) can then be used as a reference when model functions are to be fitted to the recorded currents (see below).

4.1.2. Filter Delays and Rise Times

Although the actual potential profile only deviates from the theoretical one in situations with insufficient voltage clamp control, the recorded current is generally masked by distortions from low-pass filters. The delay and the rise time of a step response are usually taken as indicative of the filter characteristics. For an eight-pole Bessel filter with a cutoff frequency f_c, the (0–10%) delay and the (10–90%) rise time are each approximately $0.34/f_c$ (see Fig. 3 and Chapter 19, this volume). If the filter characteristics are not known exactly, e.g., if the signal is filtered several times between the pipette and the actual display on the computer, it might be better to use an experimentally determined step response as a reference.

If the kinetic time constants of interest are far slower than the delays and rise times introduced by filtering, then the effects of filtering can be neglected. In several cases measured time constants or start times can be corrected simply by using the filter delay. If, for example, the sigmoidal onset of current after a voltage pulse is characterized by a delay and an exponential function to the power of *n*, the filter delay can be subtracted from the measured delay as a first approximation. For more precise analysis, theoretical functions have to be passed through an equivalent filter before they are fitted to the recorded data.

4.2. Signal Averaging and Leak Correction

4.2.1. Signal Averaging

If evoked data are recorded repeatedly under identical conditions, the current traces can be averaged in order to increase the signal-to-noise ratio, yielding smoother data traces. The statistical noise can be reduced by a factor of $\sqrt{2}$ if the number of averaged traces is doubled. Different programs use different approaches for averaging, depending on the degree of automation. These include on-line methods that acquire and average the data sweeps and only store the average, on-line methods that show the average but store all of the individual sweeps, and off-line methods that store only the individual sweeps and leave the averaging to analysis programs at a later stage. The second method is most useful, for it gives an immediate result while allowing individual records that are impeded by extraneous noise (e.g., from a current spike caused by an electrical surge) to be discarded during off-line analysis. These off-line analysis programs should facilitate data deletion and compression.

4.2.2. Leak Correction

A change in the membrane potential is accompanied by capacitive currents, which should be canceled before data analysis. This can conveniently be done when analyzing voltage-dependent conductances.

Most programs for pulsed data acquisition, therefore, support features that allow the generation of so-called *P*/*n* leak correction protocols. In a voltage range where voltage-dependent channels are not active, a scaled-down version of the pulse protocol is applied *n* times, and the resulting current is averaged, scaled, and subtracted from that elicited by the main test pulse. This method gives good results only if the signals that have to be compensated for depend linearly on voltage. In standard applications a scaling factor of $r = 0.25$ is used, and four leak responses are added to yield the scaled *P*/4 correction record (Armstrong and Bezanilla, 1974; see Fig. 8a). Because of the subtraction of a leak correction signal from the main signal, the noise is increased by a factor of $R = \sqrt{1 + 1/nr}$, with r being the ratio of leak and test pulse amplitude. For $r = 0.25$ and $n = 4$, the noise increases by $\sqrt{2}$. Leak responses should be stored together with the raw data; this ensures that one can subsequently analyze the data with and without leak correction.

Because the leak pulses can be applied from a potential other than the "normal" holding potential, a step from the holding potential to a "leak holding potential" can also create capacitive currents. These are eliminated if one performs signal averaging in which the leak

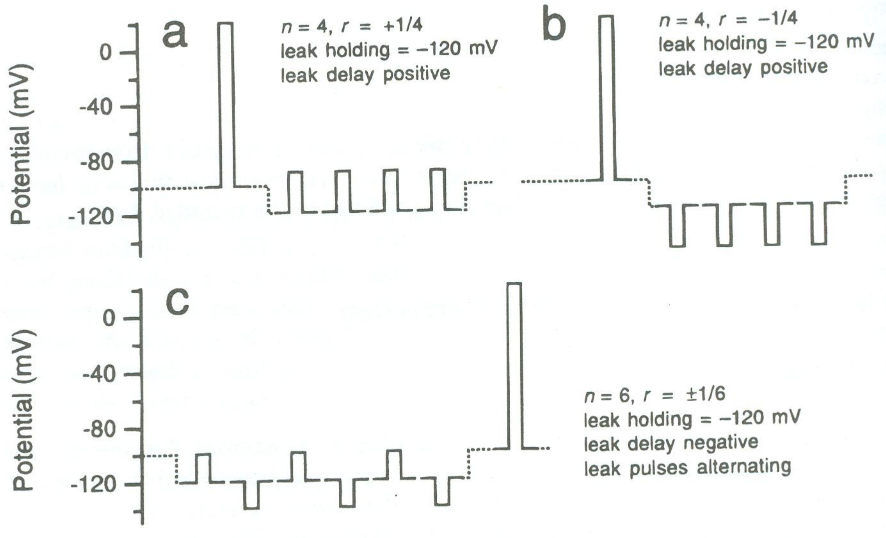

Figure 8. Pulse protocols for on-line leak correction. Panel a represents a standard *P*/4 pulse protocol with the leak pulses following the test pulse (leak delay positive). The protocol in b is similar to that in a, but the leak pulses have the opposite polarity to the test pulse. During signal averaging an alternation of the protocols shown in a and b ensures that the capacitive transients resulting from the steps from the holding potential to the leak holding potential are canceled. c: Six leak pulses, scaled by 1/6 (*P*/6 protocol), of alternating polarity precede the test pulse (leak delay negative).

pulse polarity of successive pulses alternates (Heinemann *et al.*, 1992) as illustrated in Fig. 8a,b. If no signal averaging is used, the effect of small nonlinearities around the leak holding potential can be reduced if the leak pulse polarity within a train of leak pulses is alternated (Fig. 8c). Because leak pulses might affect the currents during a test pulse or vice versa, it is best if the user has the option to record leak responses after (positive leak delay in Fig. 8) or before (negative leak delay in Fig. 8) the test pulse, respectively.

If channel activity can be abolished completely by application of a channel blocker, traces recorded after application could be used for leak correction. Analysis programs should therefore provide features to mark data sweeps to be used for leak correction of traces with channel activity.

4.3. Relaxation Experiments

For the following consideration we will assume that macroscopic currents are composed of many current events arising from the same kind of ion channel. In such cases macroscopic currents display a statistical average of many single-ion-channel events, and kinetic parameters can therefore be related to the probability functions of single-channel state transitions. The opening and closing of voltage-dependent ion channels is described theoretically in terms of kinetic Markovian schemes with channel states (e.g., open, closed, inactivated) and voltage-dependent transition rate constants (see Chapter 18, this volume). The ultimate aim of relaxation experiments is therefore to determine the state occupancies and the individual transition rates as a function of potential. Thus, pulse protocols have to be designed such that the measured macroscopic time constants can be attributed as closely as possible to microscopic time constants for state transitions.

4.3.1. Design of Pulse Patterns

A pulse pattern usually comprises a number of pulse segments of specified voltage and duration. In some cases it might be advantageous to define such segments as voltage ramps (see Fig. 6) or, for the implementation of phase-sensitive measurements, sine waves on top of a specified DC voltage (see Chapter 7, this volume). Besides a test segment, protocols usually contain at least one segment that primes the channels such that there is a defined initial condition. The kinetics of transitions among channel states are then determined.

For the investigation of time- or voltage-dependent processes either test or priming segments are varied in a systematic way, thereby creating a family of pulse patterns. From pulse to pulse individual segment durations or voltages can be altered by adding linear or exponential increments or decrements. Linear increments are widely used for segment voltages. For the investigation of steady-state inactivation properties, for example, a prepulse potential is changed, and the current is measured during a subsequent constant test pulse segment (Fig. 9c). If time constants are to be derived, e.g., the time constant for channel recovery from inactivation (Fig. 9e), exponential increments are useful, because they give rise to data points that are spaced according to their significance. Figure 9 illustrates several frequently used pulse protocols for relaxation experiments.

For analysis purposes it is quite helpful if one can specify in a pulse generator which segment is a test segment and which is a priming segment. In Fig. 9 the test segment is labeled with a "*y*" because in this segment a measurement has to be performed, and the result is to be displayed as an ordinate value. The label "*x*" denotes either a variable priming

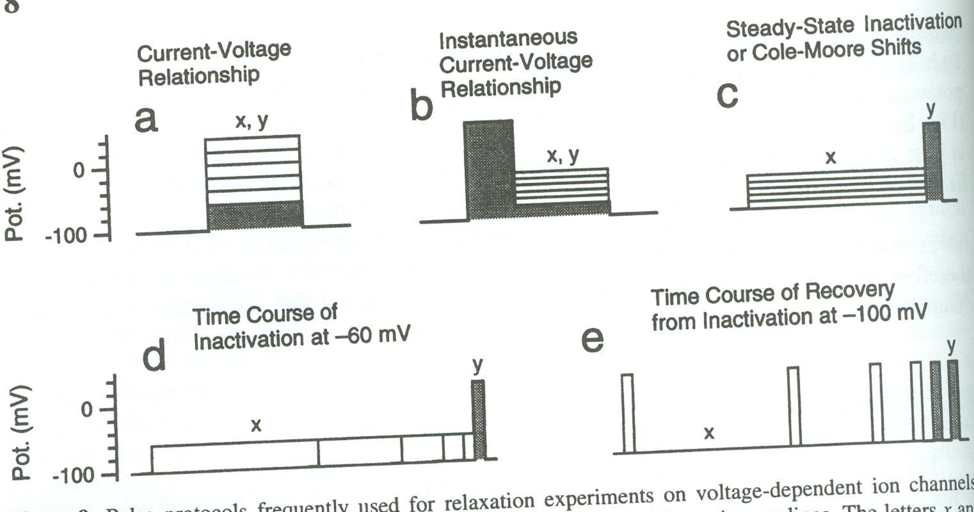

Figure 9. Pulse protocols frequently used for relaxation experiments on voltage-dependent ion channels. The first pulse is indicated by shading; successive pulses are drawn with continuous lines. The letters *x* and *y* denote which pulse segments are used as abscissa or ordinate, respectively, during secondary analysis.

segment (Fig. 9c–e) or the variable potential of the test segment (Fig. 9a,b), which are to be used as abscissa values.

4.3.2. Determination of Kinetic Parameters

Usually it is a long way from initial relaxation experiments to a kinetic scheme of channel gating. Therefore, there are several hierarchic levels for the quantitative representation of measured data and the comparison with theoretical models. Just in regard to channel activation, these levels could be as follows:

- Without any model in mind, one could describe channel activation in terms of time to peak, time to half-activation, or slope at half-activation time (see Fig. 10b). These values, determined as a function of potential, can be used as quantitative parameters to describe the channel under consideration. In further calculations these parameters can be compared with the same parameters as derived from models.

- More detailed descriptions could make use of the sums or products of several exponential functions and possibly a time delay, yielding a first estimate of how many kinetic components contribute to activation. Based on these initial guesses, analytical solutions of a devised kinetic model using idealized initial conditions can be fitted to the data. This is done during data description with Hodgkin–Huxley equations, for example (e.g., Hodgkin and Huxley, 1952), where time constants for activation are free parameters for data fit and thereby yield a more direct comparison of model and measured data (Fig. 10b).

- A more general approach is to fit all the transition rates and state occupancies of a kinetic scheme to the data directly, rather than using kinetic parameters such as relaxation constants. In this way a transition matrix (see Chapter 20, this volume) representing a particular kinetic model is fitted to entire data sweeps or even to families of sweeps without idealization of the initial conditions, yielding data descriptions that

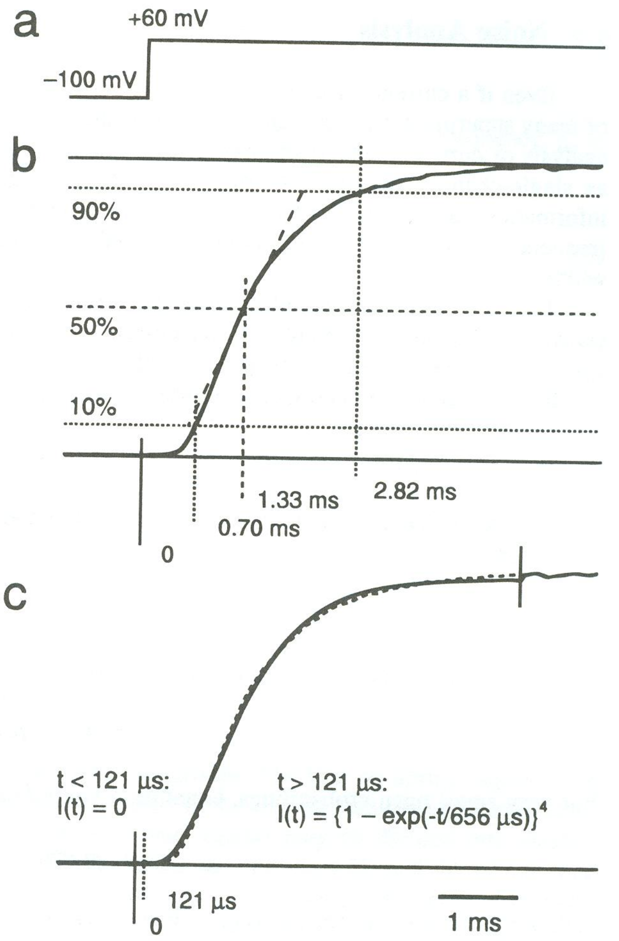

Figure 10. Examples of the description of kinetic parameters of potassium channel activation. Currents in macropatches from *Xenopus* oocytes expressing *Shaker* BΔ6-46 (Hoshi *et al.*, 1990) channels were elicited by a depolarizing step from −100 to +60 mV as shown in a. b: Activation was characterized by the time taken to reach certain current levels with respect to that taken to achieve the maximal current. Also shown is a dashed straight line through the half-maximal current. This has a slope of 55%/msec as determined by a fifth-order polynomial fit to the data in the interval between 1 and 3 msec after the beginning of the pulse. c: Same data as in b, superimposed with a fit function describing a Hodgkin–Huxley activation curve of the fourth order with an additional delay of 121 µsec. The fit was done between the two vertical solid lines. Because the data were filtered at 4 kHz with a low-pass Bessel characteristic, a filter delay of approximately 80 µsec is expected. The measured delay of 121 µsec could therefore be indicative of state transitions in addition to those described by the Hodgkin–Huxley formalism.

are not compromised by approximations. Such methods, however, are quite time consuming and are rarely implemented in standard analysis software packages.

The effects of data filtering can be considered by filtering the theoretical functions with an equivalent filter characteristic before comparison with the data. Limited clamp speed can be considered if the calculations are based on the actual potential profile rather than on the idealized voltage step (see Section 4.1.2).

Fit results derived from raw data sweeps often have to be analyzed further. Therefore analysis programs should store these results and should provide tools to display them as a function of various parameters, such as the potential of a specified pulse segment. Exponential functions, current–voltage relationships, and Boltzmann functions are often used to fit such data. For more specialized functions, tools that interpret text lines of numerical expressions (parsers) may be provided by the program; otherwise the data must be transferred to general-purpose programs (see below).

4.4. Noise Analysis

Even if a current signal recorded from a membrane patch or a whole cell is composed of many superimposed single-channel events, such that individual events cannot be detected, analysis of current fluctuations may provide information on single-channel properties such as single-channel amplitude or mean open times (e.g., Neher and Stevens, 1977). This information can be extracted from the signal either by transformation of the data into the frequency domain (Fourier analysis) or by analysis of the current variance at a given bandwidth.

In many cases valuable information can be extracted from fluctuating signals by simply considering the mean current and the current variance if the channel-open probability is small. The mean current, I, is given by the product of the single-channel current, i, the number of channels, n, and the open probability, p_o:

$$I = inp_o \tag{3}$$

Since a channel can only be open or closed, a binomial distribution applies, which has the variance:

$$\sigma^2 = i^2 np_o(1 - p_o) \tag{4}$$

With equation 3, this expression can be written as:

$$\sigma^2 = iI(1 - p_o) \tag{5}$$

For very small open probabilities, equation 5 simplifies to

$$i \approx \sigma^2/I \tag{6}$$

yielding an expression for the single-channel current.

4.4.1. Power Spectra of Continuous Data

For stationary signals, i.e., signals with a DC component that does not vary with time during an experiment, spectral analysis can be performed. For this purpose the power spectra of sections of current data are calculated and averaged. The discreteness of the signals allows the use of FFT (fast Fourier transform) algorithms, which are implemented in many software packages and can be found in the procedure libraries of development systems or statistics programs (see below). Usually the power spectra are computed from 1024 data points. The minimum and maximum frequency of the spectrum are thereby set, together with the sampling interval. If the fluctuating signal exhibits large DC components, the actual fluctuation may span only a small fraction of the input range of the AD converter. In such cases AC and DC components are sampled separately with different gain settings in order to increase the dynamic range for the fluctuating signal.

The power spectra can then be compared to theoretical predictions. Given a channel with open and closed states having exponential dwell-time distributions with the mean open and mean shut times τ_0 and τ_c, respectively, the spectral density (A^2/Hz) is described by the Lorentzian function,

$$S(f) = \frac{S(0)}{1 + (f/f_c)^2} \tag{7}$$

where $S(0)$ is the low-frequency limit of the spectral density. The cutoff frequency, f_c, is related to the relaxation time constant, $\tau = (1/\tau_o + 1/\tau_c)^{-1}$, by

$$\tau = \frac{1}{2\pi f_c}\,. \tag{8}$$

At low channel-open probability, the single-channel current amplitude, i, is obtained from the signal variance, i.e., the integral of the power spectrum [$\sigma^2 = f_c S(0)/2$], according to equation 6. The theoretical background and spectral functions for more complicated channel-gating schemes are discussed in detail by Neher and Stevens (1977) and DeFelice (1981). In particular, the existence of more than one Lorentzian component in the power spectrum can be considered as evidence of more than two distinct kinetic states. If the noise is determined by several exponential processes with very similar time constants (or a continuum of time constants), the power spectrum can acquire a shape that is not unambiguously described by a sum of Lorentzian functions. The use of power laws then helps to describe such $1/f^n$ noise.

4.4.2. Nonstationary Noise Analysis

Nonstationary signals can be divided into two groups. The first comprises signals that vary in amplitude as a function of time, e.g., as a result of the slow fluctuation in concentration of an activating transmitter at the membrane. These signals may be divided into smaller sections for determination of the time variance. When one considers the activity of only one class of ion channels, the variance is related to the DC current within the selected sections by the single-channel current and the channel-open probability. If the open probability is small, the single-channel current can be easily estimated according to equation 6.

If transient currents are recorded in response to identical repetitive stimulations, the ensemble variance can be calculated as a function of time. This is the variance caused by the deviation of each individual data point from the mean of many equivalent measurements. From equation 4 it is seen that the variance is zero if all the channels are either closed ($p_o = 0$) or open ($p_o = 1$). It reaches a maximum when half of the channels are open. If p_o is neither constant nor small, a plot of σ^2 versus I yields a parabola with a zero-crossing at the maximal current $I = i\,n$ and an initial slope corresponding to the single-channel current amplitude (Sigworth, 1980; see Fig. 11):

$$\sigma^2 = iI - I^2/n. \tag{9}$$

This method can be used for nonstationary records, where the variance is determined by computing the deviations of individual records from the mean, or, in order to eliminate drifts in the signal, from differences of successive records (e.g., Heinemann and Conti, 1992). Programs for nonstationary noise analysis must take into account background noise and leak. Automated identification of records with excess extraneous noise, based on objective statistical criteria (e.g., Heinemann and Conti, 1992), is desirable.

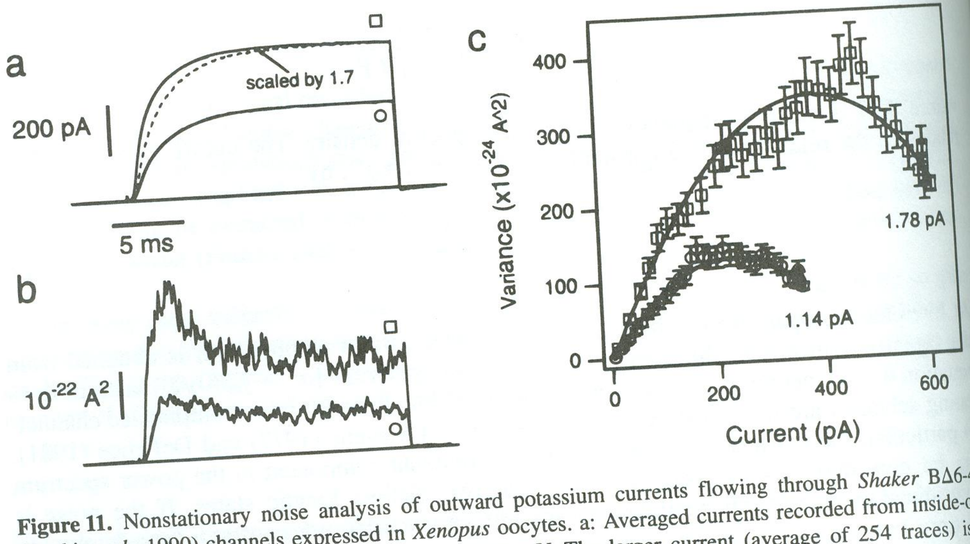

Figure 11. Nonstationary noise analysis of outward potassium currents flowing through *Shaker* BΔ6-46 (Hoshi *et al.*, 1990) channels expressed in *Xenopus* oocytes. a: Averaged currents recorded from inside-out macropatches in response to depolarizations to +80 mV. The larger current (average of 254 traces) is a control record; the smaller one (average of 356 traces) was recorded after application of sucrose to the bath, which resulted in a reduction of the current to approximately 60%. b: Averaged ensemble variance of the two experiments mentioned in a (based on 243 and 226 differences between successive current traces, respectively). c: Ensemble variance as a function of mean current. The continuous curves represent data fits according to equation 9, yielding the indicated estimates for the single-channel current and maximal channel-open probabilities of 0.78.

5. Multipurpose Programs

At many instances during execution of an experiment, during on-line analysis, or at later stages of specialized analysis, tasks arise that are readily handled by commercially available multipurpose programs. Since these programs are developed for a much larger market than patch-clamp programs, the task can be carried out with much more sophistication at a very small price. On the other hand, problems might arise during data acquisition or analysis that are so unusual that tackling them is (as yet) not implemented in such packages. In such cases one might try to extend the program that is used for acquisition and analysis (see Section 6). On many occasions, however, this is not possible, because the source code for the program might not be available, or program changes might be undesirable because even small features added to a running program may cause unwanted side effects. Alternatively, the implementation of very specialized features may simply cause an increase in complexity at the expense of user-friendliness.

In either case it is important that data and intermediate and final results can be exported from the specialized programs in such a way that they can be read by general-purpose programs for data presentation, data administration, or further analyses.

5.1. Data Presentation

Graphics programs or desktop publishing programs with graphics facilities are used to read structurally simple data files in order to generate figures for presentation. Such figures can be incorporated into text files with text-editing programs. Computer interfaces to slide writers are now widely used to generate color slides directly.

5.2. Data Administration

Data-base programs are used to generate and to manipulate complex data structures. Large data bases could, for example, be used to keep track of which channel mutants were investigated electrophysiologically, which protocols were used, and in which data files the information is stored. This will facilitate access to data files according to given experimental parameters, e.g., access to all data sweeps recorded from a certain channel mutant in a specified solution with a pulse to a specified membrane potential. For this purpose information has to be output from the acquisition or analysis program and to be interpreted by the data-base program. An alternative application is the import of information stored in a data base to the acquisition and analysis programs. The compositions of solutions used during an experiment, for example, could be imported by a dedicated analysis program to facilitate the generation of dose–response curves.

5.3. Table Calculations

Arrays of data output by acquisition or analysis programs are ideally suited to manipulation in spreadsheet programs. Such table calculation programs provide a variety of tools for data presentation and further analysis, including several statistical procedures.

5.4. Curve-Fitting Programs with Programming Capabilities

Several programs are available that support data display and the fitting of specific theoretical functions. Such specialized functions are usually composed of a set of standard functions. Alternatively, these programs provide parsers that interpret user-defined mathematical functions. Advanced programs of this kind offer their own simplified command language, which can be used to define mathematical functions or more complex analysis algorithms.

In principle, programs for data presentation, statistics, and data fitting cannot be discriminated easily because most of them offer some features for each of these purposes. However, usually these programs are particularly good for only one or two applications. One may therefore need several programs to meet all the requirements. In general, however, it should be remembered that the use of a few programs that one knows well may be better in the long run than using many. Even if these few programs do not contain all possible features, there will be fewer problems originating from data transfer, and time and money will be saved.

6. Choices for Hardware and Software

The computer market is expanding so rapidly that it is hard to keep track of all the new products and their specifications. In this section several criteria are presented that may help the reader to decide on a combination of computer, peripherals, and software that are suitable for specific experimental tasks and are compatible with the budget. The range of possible experimental tasks should be clear from the above sections and from the first chapters of this volume. The question of what should be done if the software package is insufficient in some respects will be discussed.

We will start with a description of hardware components and typical specifications that should be checked carefully before purchase. The hardware comprises the amplifier, the computer, the data storage media, and the peripherals that interface with the actual experiment, i.e., AD and DA converters. Software components such as operating systems, development software, application software, and application software with development capabilities are discussed with respect to hardware configuration. As illustrated in Fig. 12, in reality such components are not independent of each other. This means that a certain computer configuration is only compatible with some of the higher-level software available, and vice versa. During the process of evaluation of what to purchase, one will be faced with the old chicken-or-egg question. In the past, a computer and its peripherals were something precious, and software was just some imaginary quantity that could easily be copied, so the order of priority

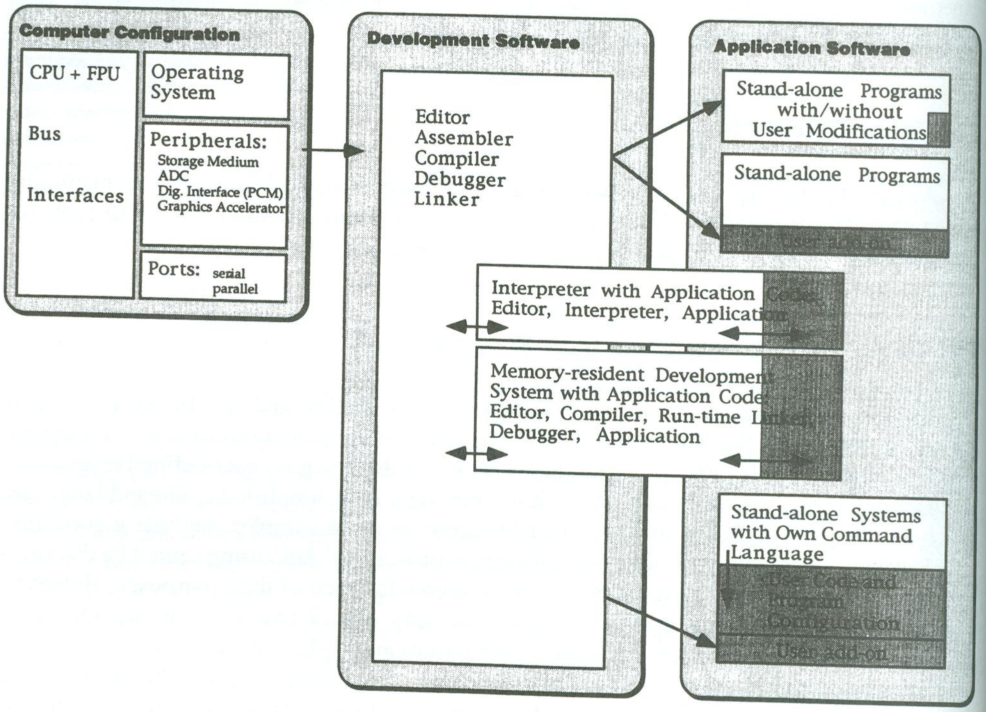

Figure 12. Organization of computer configurations and software levels. The computer hardware with all its components, including the operating system, forms a functional configuration. On top of this configuration there are roughly two more levels of software. The development software is used to generate code and to install application software, which performs such tasks as data acquisition or analysis. This separation of development and application software is only strict for compiled and linked programs that are distributed as stand-alone applications. These two kinds of software merge in so-called development systems, where development tools and application code coexist in a single running application. The arrows indicate direction of interaction. Dark shading is used to illustrate code introduced by the user. The amount of user code certainly cannot be predicted, but in general, modifications will be rare in stand-alone applications and will become mandatory in general-purpose toolbox programs that do not provide applications dedicated to electrophysiological experiments.

was hardware first and then software. However, the considerable drop in prices for computer hardware and the increased sophistication and copy protection of software tools is now clearly setting the trend for buying hardware compatible with a specified software configuration.

6.1. Criteria for the Selection of Hardware

The PDP-11 computer series (Digital Equipment Corp.) was until recently widely used for electrophysiological research applications because of its reliability and the availability of powerful real-time interfaces and programming environments. In the meantime, however, personal computers have become faster and much less expensive so that most of the applications that are now commercially available are designed to run on one of these machines. Although more sophisticated and more expensive computers, like workstations, are also used for electrophysiological experiments, most experiments are carried out using IBM-compatible and Apple Macintosh II personal computers. Therefore, only these two systems will be referred to in the following discussion.

Depending on how deeply one wants to go into the details of a computer system and its software, it can be a box with an on/off button and a stand-alone application program used for data acquisition and analysis, or it can be viewed as a highly complex modular system of hardware components and software tools that interact with each other and cannot therefore be easily considered independently of the rest. "Computer configuration" is defined here as the combination of hardware and an operating system that allows the loading of development and application software as described below.

6.1.1. Computer Configuration

The heart of a computer is its *central processing unit* (CPU), which determines speed of performance and poses the major compatibility problem among different machines. For high performance, IBM-compatible computers should be equipped with an 80486 series (Intel) processor, whereas for Macintosh computers the processor of choice is currently the 68040 CPU from Motorola. In each case the processor should be supported by a *floating point unit* (FPU), which takes over time-consuming floating point operations, thereby considerably improving performance. The type of CPU and FPU have important implications for the software that is going to run on the computer. This is the major reason for the incompatibility of applications running on IBM and Macintosh computers. A new series of PowerPCs that combine features of IBM and Macintosh computers based on a RISC (reduced instruction set computer) processor may solve some of these problems.

The CPU communicates with the environment via built-in interfaces such as serial or parallel ports. Such ports can have various specifications. Serial ports often use the RS-232 convention; a typical parallel port is an SCSI interface. Besides these standard interfaces, personal computers offer extended bus structures for the exchange of data and commands. IBM-compatible computers can have various bus structures (e.g., 16-bit AT, 32-bit EISA); for Macintosh II computers only the 32-bit NuBus is available. External devices such as AD converters, digital interfaces to a PCM, and graphics accelerators are connected to this bus. One must therefore consider the compatibility of bus and peripherals and how many bus slots are necessary.

An important determinant of computer speed and performance is the amount of disk space and the amount of *random access memory* (RAM) available. A considerable drop in the cost of both types of memory and the extended address ranges of the new generation of computers have removed some of the limitations. For a computer system that is used for real-time high-demand data acquisition, 24 Mbyte RAM and a 1 Gbyte capacity hard disk would be an appropriate configuration. Less demanding tasks can be accomplished with smaller systems.

Media for long-term data storage are hooked up either to an expansion bus or to a parallel port (e.g., SCSI). These media range from very fast removable hard disks through optical and magnetooptical devices to magnetic streamer tapes and floppy disks. The cost of all of them has come down a lot, but it is still important to consider access time and cost per megabyte before deciding which to buy. For long records of single-channel events, relatively inexpensive digital magnetic tapes, which hold several gigabytes of data, or WORM cartridges are a good choice. For data that take up less space and are not as linearly structured as single-channel records, storage media with shorter seek times and random access capabilities are desirable. For these data types removable hard disks and rewritable magnetooptical disks are appropriate.

6.1.2. Operating Systems

Operating systems are programs that manage the interactions between all components of a computer configuration, including application software, and are intimately linked to the hardware components. For personal computers there is not usually much choice of operating systems. IBM-compatible computers are equipped with MS-DOS (Microsoft disk operating system) or advanced graphics-oriented systems like *Windows* (Microsoft). The operating system of Macintosh computers is *Finder.* For both types of machine there are also other systems available (e.g., UNIX), but these are not widely used on personal computers.

6.1.3. Analogue-to-Digital Converters

Both AD and DA converters are specified by their time and voltage resolution (see Section 2.1.1). The interface between the converters and the computer bus determines the actual speed with which data can be sampled. Often AD/DA converters can sample data at a certain rate for only a short period of time, depending on the size of a buffer memory, which is then slowly emptied via the bus. In continuous recording mode the usually slower data transfer rate limits the attainable sampling rate.

A DA converter may have a problem denoted as "glitching." This results from an asynchronous switching of all relevant bits, causing the analogue output to be set briefly to an unintended value in some cases. This effect is not uncommon for DA converters and can cause serious problems if small voltage steps are distorted by short but huge voltage spikes at the transition points. Such problems are avoided by using deglitched DA converters, which, however, are more expensive. The errors caused by glitching can be reduced by (1) filtering the stimulus voltage before feeding it into the patch-clamp amplifier and (2) by the subsequent use of an optimal dynamic range for the DA conversion.

In Table I several requirements for AD/DA converters and typical specifications are summarized.

Table I. Requirements and Typical Specifications for AD/DA Converters

Requirement	Typical Specification
Number of AD and DA channels	4–8 and 2–8
Maximal sampling rate	50–200 kHz/one channel
Resolution	(8), 12, 14, 16 bits
On-board buffer	8–16k samples
DA deglitched	no/yes
Additional digital triggers (in/out)	no/yes

6.2. Criteria for the Selection of Software

Software can be grouped on several hierarchical levels. For our purposes the lowest level is the operating system (e.g., MS-DOS on IBM-compatible computers, UNIX on various workstations, or Finder on Macintosh computers). As discussed earlier, some operating systems are only available for certain computers and some application programs are only written for certain operating systems. On the next level one finds compilers/interpreters or so-called development systems. A compiler is a program that translates program code written in a specific programming language into code that can be interpreted by the computer's CPU. After this translation process the machine code is linked to operating system routines and user-defined libraries and set up such that it can be executed. High-level application programs, like those of various acquisition and analysis packages, are generated in this way. Other application software used for data analysis comprise programs for data presentation and administration, statistics, and curve fitting as outlined in Section 5.

6.2.1. Development Software

If one is not going to write one's own code for data acquisition or analysis, the steps involved in program development do not need to be considered, because commercially available acquisition and analysis programs are ready to use. If a software package that meets all requirements can be found, one should purchase it, because no special knowledge of programming is required except for the configuration of the program flow as can be determined from the program manuals. In many cases, however, such "closed" systems are not sufficient. In particular, the experienced experimenter may find it necessary to generate some extra code that complements a commercial program (e.g., by adding analysis functions or by writing extra programs in a style similar to the "master" program), or to alter the program. In some cases it may even be necessary, or at least advantageous, if one can execute such changes "on the fly," i.e., during a running experiment. In all these cases a software development system is useful because it provides an environment for writing and executing programs.

6.2.1a. ***Compiled Stand-Alone Programs.*** Figure 12 illustrates the organization of acquisition and analysis software, with potential modifications shaded. Stand-alone programs, if purchased commercially, are ready to run and cannot be modified easily. If source code, information about compilation, and linking instructions are available, the user may introduce changes, but these changes should usually be minor ones and strictly speaking should be avoided altogether because of the possibility of introducing side effects. Other stand-alone programs are provided as a set of objects (compiled code) that are linked and can be executed. These programs have built-in programming interfaces that allow user routines to be written, compiled, and linked to the system without directly interfering with the code of the main program. The user code may still cause program errors, but this method is safer than directly changing complex code that was written by someone else. For data acquisition and analysis systems, popular computer languages are C, C++, and Pascal. FORTRAN is not used much for data acquisition programs on personal computers, but, particularly for analysis purposes, it has its strengths because of the extensive subroutine libraries that are available on larger computers.

6.2.1b. ***Interpreter Systems.*** In interpreter systems program instructions are parsed, compiled, and executed line by line. Therefore, one usually has access to the source code (some parts of the programs may be supplied as object files that cannot be accessed by the user). It is quite simple then to incorporate user-specific modifications, and the program flow can be stopped and restarted at any point. A very successful product widely used in electrophysiological laboratories was the legendary Basic-23 system running on PDP-11 computers. The disadvantage of these interpreter systems is their low speed (Basic-23, therefore, made use of many subroutines written in assembly code) and the fact that the language is not very structured, leading ultimately to messy code if the program exceeds a certain size. Several interpreter systems enable the compilation and linking, once tested, of the entire code in order to generate a stand-alone program that runs faster but that can no longer be modified. This is an important option, as run time is always an issue, and a linked program is less prone to the introduction of accidental changes by inexperienced users.

6.2.1c. ***Memory-Resident Development Systems.*** On today's computers, compilation is so fast that one can afford to compile entire program modules rather than only lines. In addition, RAM has become so inexpensive that memory-resident development systems consisting of programming tools such as an editor, compiler, and debugger, which are always held in RAM, are now available. The system itself links newly compiled code to its own running application, enabling the possible immediate execution of code. Fast compilation and the essential lack of linking time makes such systems behave like a very powerful interpreter system. The advantages are clearly speed, the availability of structured programming languages suitable for large software packages, and the possibility of a high degree of user interaction. The runtime-linking, memory-resident development system *PowerMod* (HEKA Elektronik) is based on the Modula-2 programming language. Software for the control of the EPC-9 patch-clamp amplifier and the acquisition and analysis package *Pulse+-PulseFit* run in this environment.

The tool library is an important consideration if development systems are to be used for electrophysiological research. A variety of procedures operating on data arrays such as standard mathematical operations, histogram functions, or Fourier transforms are very helpful.

6.2.1d. ***Toolbox Programs.*** In contrast to development systems, which normally make use of standard programming tools such as editors, compilers, and linkers and thereby specify the use of one or several standard programming languages, some stand-alone programs provide their own comprehensive command language. This language often has a simplified structure but is supported by sets of procedures with predefined tasks. These procedures can be used as tools to set up complex programs. Efficiency is increased by the option of linking procedures written in a common programming language (e.g., Pascal or C) as additional external tools. Examples of such systems, which have been adapted to patch-clamp applications, are LabView, Igor, and ASYST.

Thus, there are a variety of options for the kind of system one should actually get. The decision as to which is most suitable will be determined by the requirements and the programming skills of the experimenter. In general, however, it should be noted that the

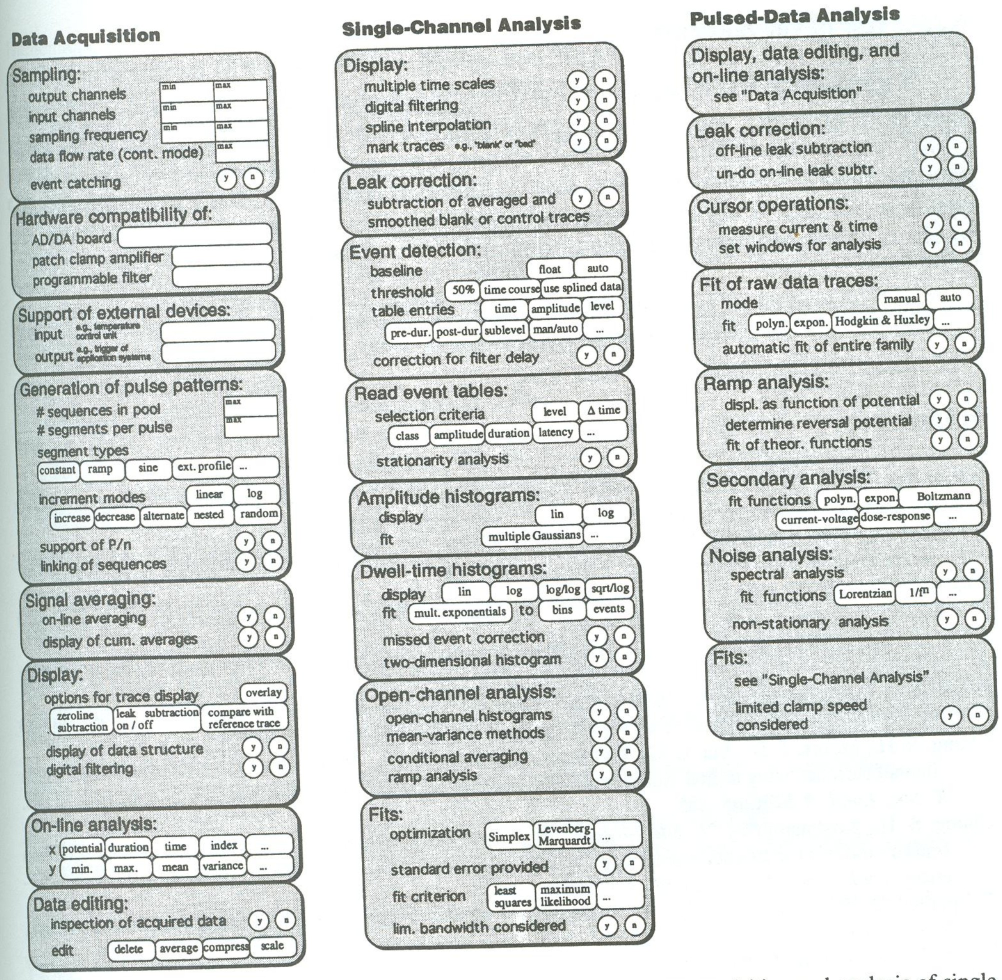

Figure 13. Checklist for essential features of software packages for data acquisition and analysis of single-channel recordings as well as pulsed data.

generation of new code is always associated with new problems and should only be considered if there is no other simpler and more economical way of solving a problem.

6.2.2. Acquisition and Analysis Systems

It is time-consuming but very important to check whether the planned experiments can actually be performed with the software under consideration. However, there is no ideal way of testing a software package other than by trying to execute an experiment with a demo version. Nevertheless, the collection of requirements and features illustrated in Fig. 13 may be used as a checklist when considering the purchase or design of acquisition or analysis software.

6.2.3. Test of the Software

A golden rule is that one should never rely blindly on software. One cannot test all the features of a complex acquisition and analysis program, but one should at least try to check critical parameters such as the voltage scaling, timing, and current gain. Errors in these parameters may indicate problems with hardware compatibility (e.g., clock rate or AD converter scaling). Analysis functions are best tested with simulated data. Alternatively, the results of the same analysis obtained using different programs can be compared.

ACKNOWLEDGMENT. I should like to thank A. Elliott and K. Gillis for their helpful comments on the manuscript and T. Schlief for his assistance in the preparation of figures.

References

Armstrong, C. M., and Bezanilla, F., 1974, Charge movement associated with the opening and closing of the activation gates of the sodium channel, *J. Gen. Physiol.* **63:**533–552.

Auerbach, A., 1993, A statistical analysis of acetylcholine receptor activation in *Xenopus* myocytes: Stepwise versus concerted models of gating, *J. Physiol.* **461:**339–378.

Bezanilla, F., 1985, A high capacity data recording device based on a digital audio processor and a video cassette recorder, *Biophys. J.* **47:**437–442.

Blatz, A. L., and Magleby, K. L., 1986, Correcting single channel data for missed events, *Biophys. J.* **49:**967–980.

Blatz, A. L., and Magleby, K. L., 1989, Adjacent interval analysis distinguishes among gating mechanisms for the fast chloride channel from rat skeletal muscle, *J. Physiol.* **410:**561–585.

Chung, S. H., Moore, J. B., Xia, L. G., Premkumar, L. S., and Gage, P. W., 1990, Characterization of single channel currents using digital signal processing techniques based on hidden Markov models, *Phil. Trans. R. Soc. Lond. B* **329:**265–285.

Chung, S. H., Krishnamurthy, V., and Moore, J. B., 1991, Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents buried in noise and deterministic interferences, *Phil. Trans. R. Soc. Lond. B* **334:**357–384.

Colquhoun, D. 1987, Practical analysis of single channel records, in: *Microelectrode Techniques. The Plymouth Workshop Handbook* (N. B. Standen, P. T. A. Gray, and M. J. Whitaker, eds.), pp. 83–104, Company of Biologists Ltd., Cambridge.

Colquhoun, D., and Sakmann, B., 1985, Fast events in single-channel currents activated by acetylcholine and its analogues at the frog muscular end-plate, *J. Physiol.* **369:**501–557.

Crouzy, S. C., and Sigworth, F. J., 1990, Yet another approach to the dwell-time omission problem of single-channel analysis, *Biophys. J.* **58:**731–743.

DeFelice, L. J., 1981, *Introduction to Membrane Noise,* Plenum Press, New York.

Dempster, J., 1993, *Computer Analysis of Electrophysiological Signals,* Academic Press, London.

Fredkin, D. R., and Rice, J. A., 1992, Maximum likelihood estimation and identification directly from single-channel recordings, *Proc. R. Soc. Lond. B* **249:**125–132.

French, R. J., and Wonderlin, W. F., 1992, Software for acquisition and analysis of ion channel data: Choices, tasks, and strategies, in: *Ion Channels, Methods in Enzymology,* Vol. 207 (B. Rudy and L. E. Iverson, eds.), pp. 711–728, Academic Press, San Diego.

Hawkes, A. G., Jalali, A., and Colquhoun, D., 1991, The distribution of open and shut times in a single channel record when brief events cannot be detected, *Phil. Trans. R. Soc. Lond. A* **332:**511–538.

Heinemann, S. H., and Conti, F., 1992, Non-stationary noise analysis and its application to patch clamp recordings, in: *Ion Channels, Methods in Enzymology,* Vol. 207 (B. Rudy and L. E. Iverson, eds.), pp. 131–148, Academic Press, San Diego.

Heinemann, S. H., and Sigworth, F. J., 1988, Open channel noise. IV. Estimation of rapid kinetics of formamide block in gramicidin A channels, *Biophys. J.* **54:**757–764.

Heinemann, S. H., and Sigworth, F. J., 1989, Estimation of Na$^+$ dwell time in the gramicidin A channel. Na ions as blockers of H$^+$ currents, *Biochim. Biophys. Acta* **987:**8–14.