Development and Application of the Original Software for Continuous High-Speed Data Acquisition of Ion Channel Current.

Yukio HOSOYA*

Abstract: The author originally has developed a program of high-speed continuous data acquisition for patch-clamp experiments. This program was designed to perform multi-input channel recording using a high-quality scientific digitizer. The source code of the program was written in C programming language and IGOR programming language. Single ventricular myocytes of a guinea pig heart were obtained by an enzymatic dissociation method. Currents of a cardiac inwardly rectifying potassium ion channel (I_{K1}) were recorded in the cell-attached configuration of the patch-clamp technique. When one input channel of the signal was acquired continuously to a hard disk, this program could acquire data at maximum sampling rate of 200 kHz without any errors. The program could take full advantages of the power and capabilities of both a digitizer and a computer. For most of the patch-clamp experiments, the obtained maximum sampling rate is high enough to perform single ion channel analysis, even in an ion channel with fast kinetics.

Key words: patch-clamp, ion channel, computer software, data acquisition

Introduction

Patch-clamp technique generally refers to both voltage-clamp and current-clamp measurements using a blunt micropipette to isolate a patch of membrane[1-3]. The patch-clamp technique is an electrophysiological method that allows the recording of macroscopic whole-cell or microscopic single ion channel currents flowing across biological membranes through ion channels. During recording, the entire pipette current flows through the membrane patch. In a single ion channel mode, patch-clamp recording measures the individual ion channel currents that contribute to the whole-cell currents. By the voltage-clamp experiment, the technique allows to experimentally control and manipulate the voltage of membrane patches or the whole cell.

In order to analyze the kinetics of an ion channel quantitatively by a computer, it is necessary to acquire the data continuously at high sampling rate[4]. The term "sampling rate" used in this paper means the rate at which the input signal is acquired on each input channel. Although there are many low-cost general-purpose digitizers and commercial programs for continuous data acquisition, only high-quality scientific digitizers from two or three companies can be used for the patch-clamp experiments[5]. Because low-cost data acquisition boards usually do not have a sufficient on-board memory to buffer data, they may not be capable of high-speed data acquisition without dropping samples[5]. In patch-clamp experiments, a high-quality scientific digitizer is required to reduce the noise level and to record signals accurately. High-quality digitizers specially designed for patch-clamping have following features[6]. All analog input

* Department of Nursing,
Yamagata School of Health Science.
260 Kamiyanagi, Yamagata, Yamagata 990-2212, Japan.

and output lines on the digitizer are optically isolated from the host computer. This optical isolation eliminates a possibility of ground loop problems arising with connected equipments. It also eliminates computer noise from analog ground.

Before 1990, there was only one commercial software package for acquisition and analysis of the patch-clamp data (pCLAMP : Axon Instruments, Inc., Foster City, CA, USA). This program supported only MS-DOS (Microsoft disk operating system : Microsoft Corporation, Redmond, WA, USA) on IBM-compatible computers. However, early version of this program could not acquire data continuously. The program read a segment of data from a digitizer, then saved these data to a hard disk. The program repeated these procedures, and recorded the data in many short segments. While the program was writing the data to a hard disk, some input signals between segments were lost, and there were short gaps between each segment. However, new improved version of this program[7] can acquire data continuously at a high sampling rate, and recently new acquisition program (Clampex 7 : Axon Instruments, Inc.)[8] for Microsoft Windows (Microsoft Corporation) was also released.

In 1992, another commercial software for data acquisition (PULSE : HEKA Elektronik, Lambrecht/Pfalz, Germany) was released. This program supported Macintosh computers (Apple Computer, Inc., Cupertino, CA, USA), and can continuously acquire the patch-clamp data. In early versions, this commercial software updated the monitor window to provide a real-time display, and the maximum sampling speed was limited. However, new version of this program[9] improved the acquisition speed, and also supported Microsoft Windows.

In order to perform high-speed continuous data acquisition, the author has written an original program for data acquisition on Macintosh computers. In 1995, the author originally developed a program for high-speed continuous data recording for patch-clamp experiments. This program was designed to perform multi-input channel recording using a high-quality scientific digitizer, and takes full advantages of the power and capabilities of both a digitizer and a computer. The program supports Apple Macintosh operating system (MacOS) running System 7.1 or later. Both PowerPC and 680x0 processors are supported.

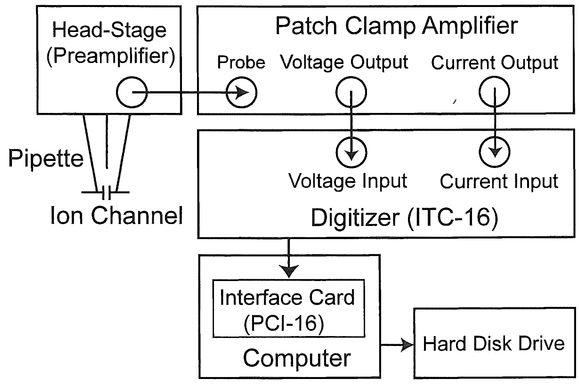

Fig. 1 Hardware of the data acquisition system.
A figure shows data acquisition hardware used in this program. Arrows in the figure indicate the direction of the signals.

Recently, another commercial program (Acquire : Bruxton Corporation, Seattle, WA, USA)[10] for continuous data acquisition was released. The program was offered for both Microsoft Windows and Apple Macintosh.

Methods

Hardware used for data acquisition

Figure 1 represents the hardware used in this acquisition program. Arrows in the figure indicate the direction of the signals. A patch-clamp amplifier[11] (EPC-7 : HEKA Elektronik) was specially designed to amplify the small signals from an ion channel. It is typically designed as two components, 1) a head-stage (pre-amplifier) mounted as close to the pipette as possible to provide initial amplification, and 2) the patch-clamp amplifier itself.

A digitizer (ITC-16 : Instrutech Corporation, Great Neck, NY, USA) converts one or more input channels of analog signals from the amplifier into a sequence of corresponding digital values. The digitizer consists of an analog to digital (A/D) converter. ITC-16 is a 16-bit digitizer with a maximum sampling rate of 200 kHz[6]. Eight analog input channels are available, and multiplexed to a single A/D converter. The digitizer has 16K samples of A/D first-in first-out (FIFO) memory. In this

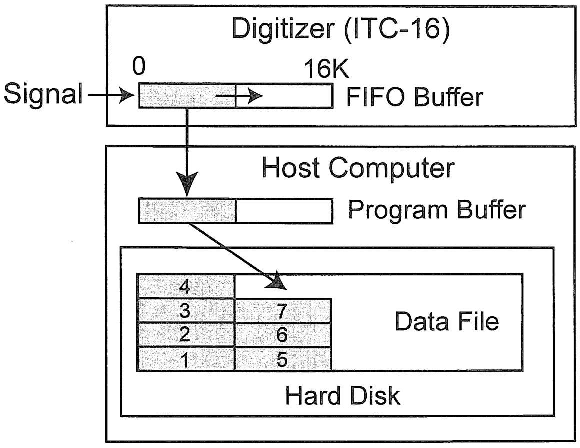

Fig. 2 Simplified data flow in the acquisition program. Shaded box represents a block of data to be transferred. A block of data read from the FIFO Buffer of a digitizer is stored temporarily in the Program Buffer before further processing. Next, the program reads a block of data out of the Program Buffer periodically and writes and appends this block of data into the output file on the storage device periodically. The number in the shaded box represents the order of the data transfered to the output data file, and this number indicates the segment number of the data.
FIFO : first-in first-out, 16K : 16 × 1024 samples.

paper, 1K means 1,024 (i.e. 2^{10}), while 1k means 1,000 (i.e. 10^3). ITC-16 is an external data acquisition unit, and is connected to the host computer using an interface board. An interface card (PCI-16 : Instrutech Corp.)[12] is a PCI (peripheral component interconnect) bus plug in card for computers. PCI-16 card provides the necessary interfacing to connect from a digitizer (ITC-16) to a host computer (Power Macintosh 7600/200 : Apple computer, Inc.).

Data flow in the acquisition program

Figure 2 illustrates the simplified data flow used in this program. Data are digitized by an A/D converter, and transferred to a computer's memory. Digitizers used for high-speed measurement can feed data to a computer at a high and constant rate. For example, a digitizer running on one input channel at 100K samples/sec will produce 200K bytes/sec of data continuously. The continuous nature of such data acquisition requires a buffering memory. For example, if the computer stops for 40 msec to write data to disk or to update a display, 8,192 bytes of data will accumulate. These data must be stored somewhere, or it will be lost.

ITC-16 has a 16K sample FIFO buffer. FIFO is a memory used to buffer input data. Input data acquired by ITC-16 is transformed to this FIFO memory, and can be read by the acquisition program. ITC-16 uses the FIFO buffer to hold data, until it can be processed by the acquisition program. The data is then transferred to a host computer by programmed Input/Output (I/O) system. The acquisition program performs this transfer. The Instrutech digitizers do not provide interrupts to the host computer. Instead, host computer must periodically poll the device to obtain data. This polling is performed periodically by the application program.

For the continuous sampling, the program must start acquisition and then monitor the counters showing the available bytes and the data overrun flags. At high sampling rate, it becomes more critical to keep the FIFO sampling buffer as empty as possible. For example, if acquisition was performed at a fast sampling period of 10 μsec/sample (i.e. sampling rate is equal to 100 kHz), and the FIFO sample buffer is 16K long, that leaves only 163.84 msec before the FIFO overruns. The acquisition program should therefore read the FIFO into program's own buffer about every 100 msec in this example.

Algorithm of the acquisition program

Figure 3 represents a simplified main algorithm of this program. Since an ITC-16 has relatively small buffer memory (16K samples), it is difficult to develop a computer program for high-speed continuous data acquisition on a multitasking operation system. This program is capable of full-rate continuous data acquisition, because the program does not display the incoming data during recording. The program is designed to move data quickly from the digitizer to a storage device.

The user precisely defines the exact length and occurrence of data acquisition. The program initializes a digitizer, and then creates a program's own buffer memory (referred to in this paper as "Program Buffer") in the RAM (random access memory) of the host computer. The program creates and opens an output data file on the storage device

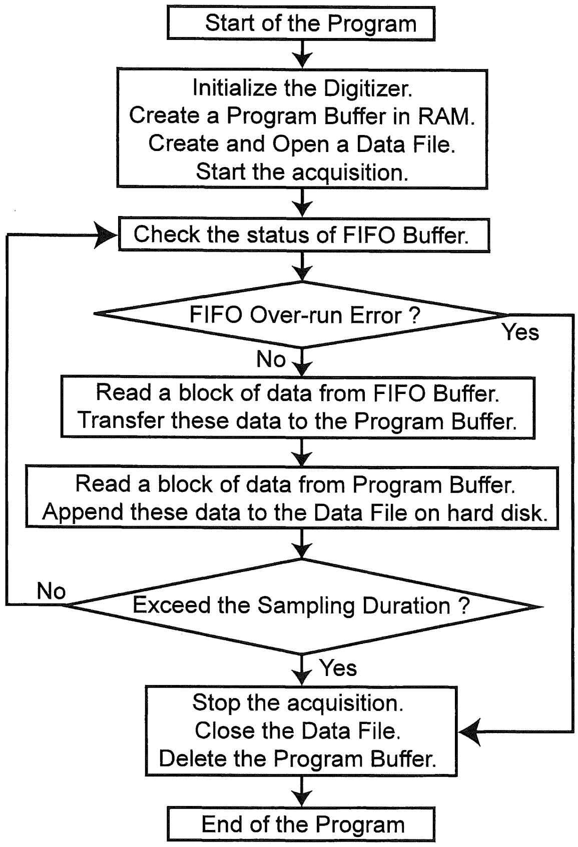

Fig. 3 Algorithm of the acquisition program.
A figure represents simplified main algorithm used in this program.
FIFO : first-in first-out, RAM : random access memory.

(e.g. hard disk drive or magneto-optical (MO) disk drive), and then starts the acquisition. The acquisition device has an internal buffer (FIFO Buffer). The program reads a block of data out of the FIFO Buffer of the acquisition device periodically and writes this data into the Program Buffer (see also Fig. 2). The program reads data out of the digitizer before the FIFO Buffer overflows. The program calculates the optimum time interval to use to empty the internal buffer of the acquisition device.

A block of data read from the FIFO Buffer of the digitizer is stored temporarily in the Program Buffer before further processing. Next, the program reads a block of data out of the Program Buffer periodically and writes and appends this block of data into the output file on the storage device periodically. The program checks the status of FIFO Buffer periodically. If the FIFO overrun error occurs, the program stops the acquisition, and displays an error massage. These procedures were repeated until the recording duration exceeds the value specified by the user, or until the user stops the acquisition during recording. The program stops the acquisition, closes the data file, and deletes the Program Buffer.

The program acquires the input signal and directly records the arbitrary long data file to disk. Data are streamed non-stop to disk. There are no gaps in the data file, because every single data segment is written to disk, up to the limit of the disk capacity. Writing to the disk does not introduce gaps into the data. This gap-free recording mode is suitable for recording single ion channel current in order to study the kinetics of the ion channel.

Development of the program

A software package for data acquisition was originally produced by the use of ITC-16. The acquisition and analysis system was built using IGOR Pro[13] (WaveMetrics, Inc., Lake Oswego, OR, USA). IGOR Pro is an extensible graphing, data analysis, and programming tool for scientists and engineers. IGOR Pro is an integrated application program for visualizing, analyzing, transforming, and presenting data. IGOR's features include 1) the automation and data processing via a built-in programming environment, and 2) the extensibility through modules written in the C programming language.

Programmers can use IGOR Pro as an all-purpose workhouse to acquire, analyze, and present experimental data using its built-in programming languages[14]. The source codes used for the acquisition program consisted of two components : 1) data acquisition program which is the core of this software, written in the C language, and 2) programs for user-interface, displaying oscilloscope window, and creating publication-quality graphs of the acquired traces, which were written in the IGOR's programming languages.

The source codes of user functions and macros were written using the IGOR's programming language. The source codes of the functions and macros consisted of about 2,000 lines of codes. Macros were used for user-interface part of the programming, for example, displaying dialog boxes to prompt for input values. Functions compiled by IGOR Pro were used for number crunching, especially for, procedures necessary to iterate through each point in a wave. Compilation consists of analyzing the text of a function and producing low-level instructions that can be executed quickly later.

Direct access to the acquisition device was done by creating an XOP (external operation) file, which was written specifically for ITC-16. The IGOR XOP Toolkit[15] is an add-on package that allows C and C++ programmers to extend IGOR Pro. An XOP is a relatively small piece of code which extends IGOR Pro. Although "XOP" literally means "external operation", it has also the meaning "external module that extends IGOR Pro". An elaborate XOP can add multiple operations, functions, menu items, dialogs and windows. It is even possible to build a data acquisition and analysis system on the top of IGOR Pro.

Source codes of the program for continuous data acquisition were written in the C programming language. The source codes consisted of about 1,600 lines of C codes. The routines illustrated in figure 3 were all written in the C language, because these routines should be executed as fast as possible in order to perform high-speed continuous acquisition. These source files were compiled along with the source files supplied by IGOR XOP Toolkit (WaveMetrics, Inc.), under the development system (CodeWarrior CW10 : Metrowerks, Inc., St-Laurent QC, Canada). The CodeWarrior[16] is a C and C++ development system for 680x0 and PowerPC Macintoshes. A compiler is the software that translates a program code written in a specific programming language (e.g. C language) into a machine code that can be directly executed by the computer's CPU (central processing unit).

The compiled code was then linked with the library files from IGOR XOP Toolkit, and ITC-16 device drivers[17] (Bruxton Corporation), and the executable XOP file for data acquisition was produced. Because the XOP file was compiled to the machine code, it runs faster than IGOR's built-in programming language, and can directly access the Macintosh operating system calls (e.g. Macintosh

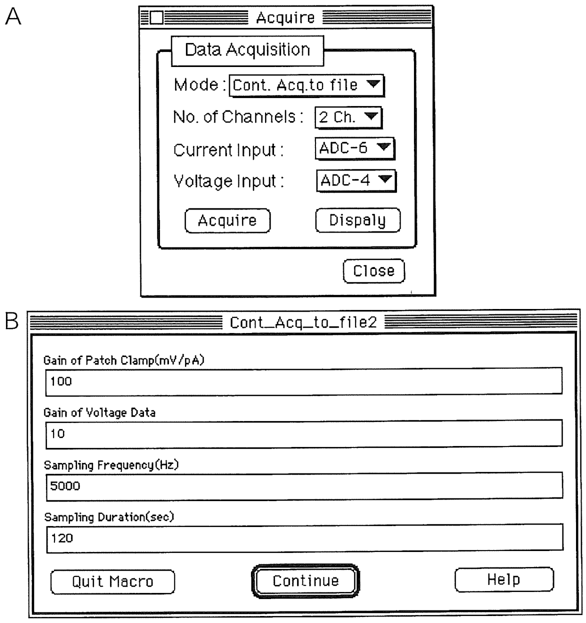

Fig. 4 Front panel (A) and dialog box (B) for data acquisition.
Users can select acquisition modes and input values using the panel and dialog box.
Cont. Acq. : continuous acquisition, Ch. : channels.
ADC : analog to digital converter.

Toolboxes) and control the hardware device via the device drivers.

Methods of the patch-clamp experiments

Single ventricular myocytes of the guinea pig heart were obtained by the enzymatic dissociation method described previously[18-24]. The heart was retrogradely perfused through the coronary arteries on a Langendorff apparatus with nominally Ca^{2+} free solution containing collagenase. To isolate single ventricular cell, a small piece of ventricular tissue was dissected and agitated in the recording chamber. Currents of a cardiac inwardly rectifying potassium ion channel (I_{K1}) were recorded in the cell-attached configuration of the patch-clamp technique[2].

Results

Description of the acquisition program

Figure 4A represents an instrument-like front panel for the data acquisition setup, using some pop-up menus and buttons. Users can select acquisition modes (e.g. continuous acquisition or triggered acquisition), storage devices (e.g. hard disk drives, or RAM), number of input channels acquired, and channel numbers of ADC (analog to digital converter)

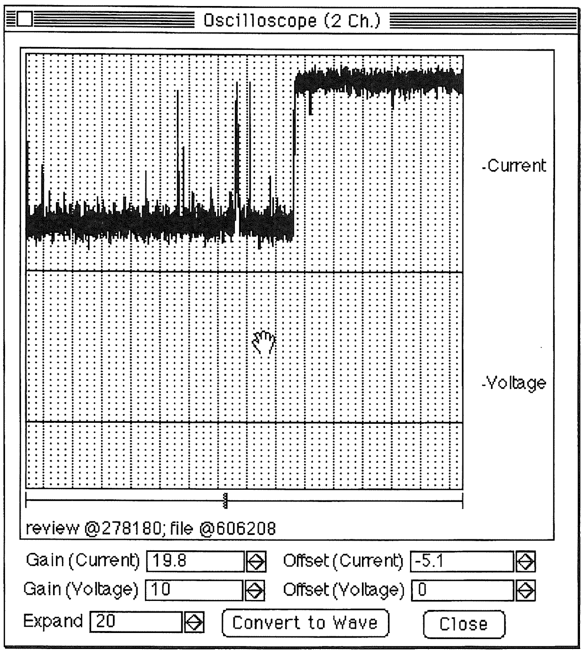

Fig. 5 Oscilloscope window.
Users can review the acquired data using this window. The upper trace represents the current of a cardiac inwardly rectifying potassium ion channel (I_{K1}), while the lower trace shows the holding potential in the voltage-clamp experiment.

for the current or voltage input. When the user clicks "Acquire" button, a dialog box will be displayed as shown in figure 4B. The user can input the gains of current and voltage signals, sampling frequency, and sampling duration. If the user clicks "Continue" button, the acquisition will start.

While the acquisition is in progress, this program buffers the data coming from data acquisition hardware, and also writes the data to an output file (Fig. 2). This gap-free recording mode records the data continuously to disk without loss of the data (without gaps). The user can record the data for a long time, up to the capacity of the disk. Maximum sampling speed depends on the computer, hard disk, and interface.

When data acquisition is finished, the process can be reversed with data coming back out of the file, and the oscilloscope window will be displayed as shown in figure 5. This window can be used to review a long data acquired previously. The upper trace represents the current of a cardiac inwardly rectifying potassium ion channel (I_{K1}), while the lower trace shows the holding potential used in the voltage-

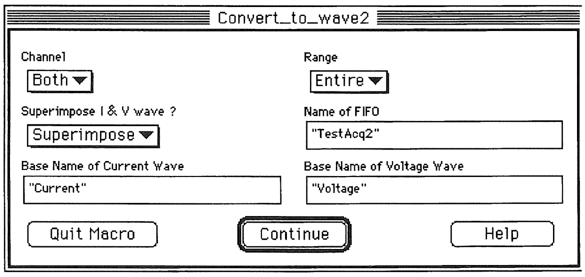

Fig. 6 Dialog box for displaying the acquired signals. Users can select the ways to display the traces.

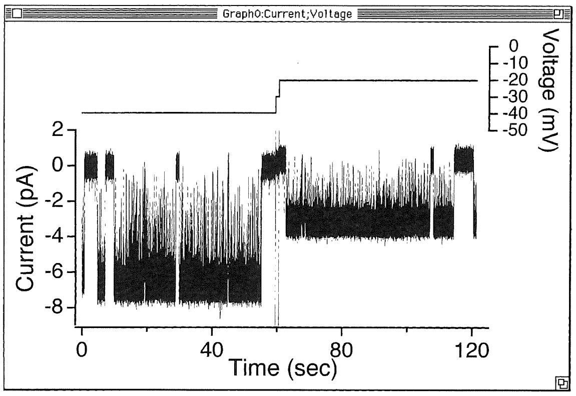

Fig. 7 Window displaying the whole traces.
The program can create publication-quality graphs of the acquired traces. The upper trace represents the holding potential, while the lower trace shows the current of I_{K1}.

clamp experiment. Users can change the gain and offset of each input channel. This window emulates a mechanical chart recorder to write on a paper with moving pens as paper scrolls under the pens. When the user positions the mouse over the oscilloscope area, the mouse cursor turns into a "hand". The user can move the chart "paper" right to left by dragging with the "hand". If the user gives the paper a push, it will continue scrolling until it hits the end. This allows the user to review the contents of a data file by scrolling the chart "paper" back and forth. The horizontal line under the scrolling paper area represents the all of the data in the output file, while the vertical gray bar represents the position of the data that is being shown in the chart.

If the user clicks "Convert to wave" button, a dialog box will be displayed as shown in figure 6. In this dialog box, the user can select the ways to display traces, and can input a name of the each trace. If the user clicks "Continue" button, a window showing whole traces will be displayed as represented in figure

Table 1 Maximum sampling rate

	Storage Device	
	RAM	Hard Disk
1 input channel	200kHz	200kHz
2 input channels	100kHz	100kHz

RAM : random access memory

7. The program can create publication-quality graphs of the acquired traces. The upper trace represents the holding potential, while the lower trace shows the current of the potassium ion channel (I_{K1}). The author has made a software for single ion channel analysis and whole cell analysis using IGOR and C programming languages. A user can analyze the data using this analysis program.

Maximum acquisition rate

Table 1 summarizes the result of the maximum acquisition rate achieved by this acquisition program. In this experiment, Power Macintosh 7600/200 (with 200 MHz PowerPC 604 CPU, 64 Mbytes RAM, and 2 Gbytes hard disk drive) was used for a host computer. Disk cache size was set to 32 Kbytes in the experiment. When only one input channel of signal (i.e. current data) was acquired continuously to the RAM, the maximum sampling rate was 200 kHz, which is the hardware limit of ITC-16[6]. When one input channel of signal was acquired continuously to a hard disk, the program also could acquire the data at maximum sampling rate of 200 kHz without any errors.

If two input channels of the signals (i.e. both current and voltage data) were acquired continuously to a RAM, the maximum sampling rate was 100 kHz for each input channel, which is equal to the limit of ITC-16[6]. When two input channels of signals were acquired continuously to a hard disk, the maximum sampling rate was also 100 kHz for each input channel. The program could take full advantages of the power and capabilities of both a digitizer and a computer.

The maximum sampling rate depends on the performance of the CPU, Bus speed, access time of RAM, disk cache size, and a maximum transfer rate of a hard disk drive. Now much faster computers are available, however computers with average speed are already enough to do high-speed continuous acquisition by using this program.

Discussion

Sampling rate

The purpose of data acquisition is to analyze an analog signal in digital form. For this purpose, the sequence of values produced by a digitizer must represent the original analog signal. The sampling theorem (i.e. Nyquist theorem) states that the sampling rate must be at least twice the signal bandwidth in order to reconstruct the signal[25, 26]. For example, if signals of an ion channel contain frequencies from DC (0Hz) to 10 kHz, this signal must be sampled at a rate of at least 20 kHz (i.e. Nyquist rate) to be reconstructed properly. Although the Nyquist theorem requires sampling at only twice the highest frequency contained in the signal, input data should be sampled ordinarily at several times the Nyquist rate to avoid aliasing signals[27]. As a practical matter, a sampling rate of 50 kHz or more should be used in this example.

The accuracy of the approximation of the input signal depends upon the sampling rate. The greater the sampling rate, the better the approximation of the signal. High-speed acquisition allows studies on the fast kinetics of ion channels. By using this program, the user can acquire data continuously at high sampling rate for long duration. The maximum acquisition rate is limited by the hardware in the several ways. The processor speed of the computer is important when processing data from the data acquisition device. The I/O speed is also important when transferring the acquired data to a disk. The maximum sampling rates of this program are high enough to perform single ion channel analysis even for an ion channel with fast kinetics.

Development of the acquisition program

In general, it is difficult to write a programming code for continuous high-speed data acquisition. To write elaborate software, one needs to be an experienced programmer. One needs to be very familiar with the programming language (e.g. C language), development system (e.g. C compiler), operating system (MacOS) including the Macintosh toolbox, and the particular hardware (ITC-16, and PCI-16).

Although several software packages for data acquisition and analysis are commercially available, such "closed" programs are not sufficient in many cases. The commercial software cannot be modified easily, because generally software companies do not supply the source code of the commercial program to the users. On the other hand, it is easy to change the program with original software. One can add some analysis functions or can alter the program for the particular experiments. In some cases it may even be necessary, or at least advantageous to execute such changes for each experiment.

Limitations and further improvements

Most commercial acquisition software can display incoming data in the virtual oscilloscope window, which provides an oscilloscope-like display of the acquired data. While data acquisition is in progress, the window is updated at regular intervals. In contrast, the present program cannot support a real-time display during data acquisition, because the purpose of this program is to increase the maximum acquisition speed. Users can review the acquired data file, only after acquisition is finished. However, this limitation is not a critical problem. Because most patch-clamp setups are equipped with a real oscilloscope and a chart recorder, the user can observe the signals on a real oscilloscope or chart recorder rather than from the computer display during acquisition. Even in the commercial software packages, the real-time display of the acquiring data might often be quite sluggish and more difficult to scale appropriately. Therefore, a real oscilloscope or chart recorder is also required, even with the commercial programs.

This acquisition software should be improved. The present program supports only particular digitizer (ITC-16) and operating system (Mac OS). In the near future, the program will be able to acquire data using a new digitizer (ITC-18 [28] : Instrutech Corp.), and also support other platform (MS-Windows computer). This acquisition program was specifically oriented towards patch-clamping, but may be useful for anyone to record multi-channel signals at a high sampling rate. By modifying this software, it is possible to develop a new program to acquire and analyze other biological signals, and contribute to other research area.

References

1) Neher E., Sakmann B. : Single channel currents recorded from membrane of denervated frog muscle fibers. *Nature* 260 : 779-802, 1976.

2) Hamil O.P., Marty A., Neher E., Sakmann B., Sigworth F.J. : Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches. *Pflügers Archiv : European Journal of Physiology* 391 : 85-100, 1981.

3) Penner R. : A practical guide to patch clamping. In "Single-Channel Recording, Second Edition", Sakmann B., Neher E. (Eds.), New York, Plenum Publishing Corporation, 1995.

4) Heinemann S.H. : Guide to data acquisition and analysis. In "Single-Channel Recording, Second Edition" , Sakmann B., Neher E. (Eds.), New York, Plenum Publishing Corporation, 1995.

5) Bruxton Corporation : A short introduction to data acquisition, with particular emphasis on the hardware and software issues involved in high-quality recording. Monographs by Bruxton Corporation, 1998.

6) ITC-16 Computer Interface, User's Manual, Instrutech Corporation, 1998.

7) pCLAMP 6 User's Guide : Axon Instruments, Inc., 1993.

8) Clampex 7 User's Guide : Axon Instruments, Inc., 1998.

9) PULSE/PULSEFIT User's Manual : HEKA Elektronik, 1999.

10) Acquire Ver.3.0 User's Manual : Bruxton Corporation, 1997.

11) Sigworth F.J. : EPC-7 User's Manual, List-electronic, 1994.

12) PCI-16 PCI Bus Host Interface Card, User's

Manual, Instrutech Corporation, 1998.

13) IGOR Pro Version 3.1, User's Guide, WaveMetrics, Inc., 1999.

14) IGOR Pro Version 3.1, Programming and Reference Manual, WaveMetrics, Inc., 1999.

15) IGOR XOP Toolkit Version 3, Reference Manual, WaveMetrics, Inc., 1996.

16) CodeWarrior CW10 Version 1.7, User's Guide, Metrowerks, Inc., 1996.

17) ITC-16/18 Driver Manual, Bruxton Corporation, 1998.

18) Yamada M., Jahangir A., Hosoya Y., Inanobe A., Katada T., Kurachi Y. : G_K* and brain $G_{\beta\gamma}$ activate muscarinic K^+ channel through the same mechanism. *Journal of Biological Chemistry* 268 : 24551-24554, 1993.

19) Terzic A., Findlay I., Hosoya Y., Kurachi Y.: Dualistic behavior of ATP-sensitive K^+ channels towards intracellular nucleoside diphosphates. *Neuron* 12 : 1049-1058, 1994.

20) Ito H., Hosoya Y., Inanobe A., Tomoike H., Endoh M. : Acetylcholine and adenosine activate the G protein-gated muscarinic K^+ channel in ferret ventricular myocytes. *Naunyn-Schemiedeberg's Archives of Pharmacology* 351 : 610-617, 1995.

21) Hosoya Y., Yamada M., Ito H., Kurachi Y. : A functional model for G protein activation of the muscarinic K^+ channel in guinea pig atrial myocytes : Spectral analysis of the effect of GTP on single-channel kinetics. *Journal of General Physiology* 108 : 485-495, 1996.

22) Hosoya Y., Kurachi Y. : Functional analyses of G-protein activation of cardiac K_G channel. In "Potassium Ion Channels : Molecular Structure, Function, and Diseases" *Current Topics in Membranes* 46 : 355-369, San Diego, Academic Press, 1999.

23) Kurachi Y., Nakajima T., Sugimoto T. : On the mechanism of activation of muscarinic K^+ channels by adenosine in isolated atrial cells : involvement of GTP-binding proteins. *Pflügers Archiv : European Journal of Physiology* 407 : 264-274, 1986.

24) Yamada M., Terzic A., Kurachi Y. : Regulation of potassium channels by G-protein subunits and arachidonic acid metabolites. In Methods in Enzymology : Heterotrimeric G-protein Effectors, Vol.238 : 394-422, San Diego, Academic Press, 1994.

25) Oppenheim A.V., Schafer R.W. : Digital Signal Processing, New Jersey, Prentice-Hall, Englewood Cliffs, 1975.

26) Crochiere R.E., Rabiner L.R. : Multirate Digital Signal Processing, New Jersey, Prentice-Hall, Englewood Cliffs, 1983.

27) Horowitz P., Hill W. : The Art of Electronics, Second Edition, Cambridge, Cambridge University Press, 1989.

28) ITC-18 Computer Interface, User's Manual, Instrutech Corporation, 1999.

Received : October 31, 1999.
Accepted : February 2, 2000.