pCLAMP 10

Data Acquisition and Analysis For Comprehensive Electrophysiology

User Guide

Molecular Devices

Molecular Devices Corporation
1311 Orleans Drive Sunnyvale, California 94089
Part #1-2500-0180 Rev. A.

Copyright

Disclaimer

Molecular Devices Corporation reserves the right to change its products and services at any time to incorporate technological developments. This user guide is subject to change without notice.

Although this user guide has been prepared with every precaution to ensure accuracy, Molecular Devices Corporation assumes no liability for any errors or omissions, nor for any damages resulting from the application or use of this information.

Questions?

Phone:	1 (800) 635-5577
Fax:	+1 (510) 675-6300
Web:	www.moleculardevices.com

Customer License Agreement

Customer License Agreement for Single User of pCLAMP

This software is licensed by Molecular Devices Corp. ("MDC") to you for use on the terms set forth below. By opening the sealed software package, and/or by using the software, you agree to be bound by the terms of this agreement.

MDC hereby agrees to grant you a nonexclusive license to use the enclosed MDC software (the "SOFTWARE") subject to the terms and restrictions set forth in this License Agreement.

Copyright

The SOFTWARE and its documentation are owned by MDC and are protected by United States copyright laws and international treaty provisions. This SOFTWARE may not be copied for resale or for bundling with other products without prior written permission from MDC.

Restrictions on Use and Transfer

You may not reverse engineer, decompile, disassemble, or create derivative works from the SOFTWARE.

Export of Software

You agree not to export the SOFTWARE in violation of any United States statute or regulation.

Ownership of Software and Media (CD-ROM)

You own the media (CD-ROM) on which the SOFTWARE is recorded, but MDC owns the SOFTWARE and all copies of the SOFTWARE.

Product Improvements

MDC reserves the right to make corrections or improvements to the SOFTWARE and its documentation and to the related media at any time without notice, and with no responsibility to provide these changes to purchasers of earlier versions of such products.

Term

This license is effective until terminated. You may terminate it by destroying the SOFTWARE and its documentation and all copies thereof. This license will also terminate if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy all copies of the SOFTWARE and its documentation.

Limited Warranty and Disclaimer of Liability

MDC warrants that the media on which the SOFTWARE is recorded and the documentation provided with the SOFTWARE are free from defects in materials and workmanship under normal use. For 90 days from the date of receipt, MDC will repair or replace without cost to you any defective products returned to the factory properly packaged with transportation charges prepaid. MDC will pay for the return of the product to you, but if the return shipment is to a location outside the United States, you will be responsible for paying all duties and taxes.

Before returning defective products to the factory, you must contact MDC to obtain a Service Request (SR) number and shipping instructions. Failure to do so will cause long delays and additional expense to you.

MDC has no control over your use of the SOFTWARE. Therefore, MDC does not, and cannot, warrant the results or performance that may be obtained by its use. The entire risk as to the results and performance of the SOFTWARE is assumed by you. Should the SOFTWARE or its documentation prove defective, you assume the entire cost of all necessary servicing, repair or correction. Neither MDC nor anyone else who has been involved in the creation, production, or delivery of this SOFTWARE and its documentation shall be liable for any direct, indirect, consequential, or incidental damages arising out of the use or inability to use such products, even if MDC has been advised of the possibility of such damages or claim.

This warranty is in lieu of all other warranties, expressed or implied. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitations or exclusions may not apply to you.

U.S. Government Restricted Rights

The SOFTWARE and its documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software -Restricted Rights at 48 CFR 52.227-19, or clause 18-52.227-86(d) of the NASA Supplement to the FAR, as applicable. Manufacturer is Molecular Devices Corp., Sunnyvale, CA 94089 USA.

Governing Body

This Agreement is governed by the laws of the State of California.

Contents

1. Introduction

New Features 1
AxoScope 2
MiniDigi 1 2
pCLAMP Documentation 3
Overview of User Guide 5
Utility Programs 6
History of pCLAMP 6

2. Description

Definitions 7
Data Acquisition Modes 10
Terms and Conventions in Electrophysiology 11
The Sampling Theorem in Clampfit 15
Optimal Data Acquisition 16
File Formats 18
pCLAMP Quantitative Limits 21

3. Setup

Computer System 25
Software Setup and Installation 27
Digitizer Configuration in Clampex 28
MiniDigi Installation 29
Resetting Program Defaults 30
Printing 30

4. Clampex Features

Clampex Windows 31
Telegraphs 34
Lab Bench 35

Overrides. 36

Handling Data 36

Protocol Editor 37

Data Acquisition 43

Real Time Controls 43

Seal and Cell Quality: Membrane Test. 44

Time, Comment, and Voice Tags. 47

Junction Potential Calculator. 48

Calibration Wizard 48

Sequencing Keys 49

LTP Assistant. 49

5. Clampex Tutorials

I-V Tutorial 65

Membrane Test Tutorial. 77

Scenarios 79

6. Clampfit Features

Clampfit Windows 85

File Import 90

Data Conditioning 90

Event Detection. 92

Single-Channel Analysis in Clampfit 94

Fitting and Statistical Analysis 104

Creating Figures in the Layout Window 105

7. Clampfit Tutorials

Creating Quick Graphs 107

Preconditioning Noisy Single-Channel Recordings 112

Evaluation of Multicomponent Signals: Sensillar Potentials with Superimposed Action Potentials 116

Separating Action Potentials by their Shape. 123

8. Digital Filters

Finite vs. Infinite Impulse Response Filters 131

Digital Filter Characteristics 133

End Effects 134
Bessel Lowpass Filter (8 Pole) Specifications 134
Boxcar Smoothing Filter Specifications 136
Butterworth Lowpass Filter (8 Pole) Specifications 137
Chebyshev Lowpass Filter (8 Pole) Specifications. 139
Gaussian Lowpass Filter Specifications. 141
Notch Filter (2 Pole) Specifications 142
RC Lowpass Filter (single Pole) Specifications 143
RC Lowpass Filter (8 Pole) Specifications 145
RC Highpass Filter (Single Pole) Specifications 147
Bessel Highpass Filter (8-Pole Analog) Specifications 147
The Electrical Interference Filter 148

9. Clampfit Analysis

The Fourier Series 157
The Fourier Transform 158
The Fast Fourier Transform 159
The Power Spectrum 159
Limitations 160
Windowing 160
Segment Overlapping 161
Transform Length vs. Display Resolution 161

10. pCLAMP Analyses

Membrane Test 163
Template Matching 167
Single-Channel Event Amplitudes 167
Level Updating in Single-Channel Searches 168
Kolmogorov-Smirnov Test 169
Normalization Functions. 170
Variance-Mean (V-M) Analysis 172
Burst Analysis 174
Peri-event Analysis. 175
$P_{(open)}$ 176

11. Curve Fitting

Introduction 179
The Levenberg-Marquardt Method 182
The Simplex Method 184
The Variable Metric Method 186
The Chebyshev Transform 187
Maximum Likelihood Estimation 201
Model Comparison 203
Defining a Custom Function 205
Multiple-Term Fitting Models 206
Minimization Functions 206
Weighting 207
Normalized Proportions 209
Zero-shifting 209

12. Fitting Functions

Beta Function 211
Binomial 212
Boltzmann, Charge-Voltage 212
Boltzmann, Shifted 213
Boltzmann, Standard 213
Boltzmann, Z-delta 214
Current-Time Course (Hodgkin-Huxley) 214
Exponential, Alpha 215
Exponential, Cumulative Probability 215
Exponential, Log Probability 215
Exponential, Power 216
Exponential, Probability 216
Exponential, Product 216
Exponential, Sloping Baseline 216
Exponential, Standard 217
Exponential, Weighted 217
Exponential, Weighted/Constrained 217
Gaussian 218
Goldman-Hodgkin-Katz 218

Goldman-Hodgkin-Katz, Extended 218
Hill (4-Parameter Logistic) 219
Hill, Langmuir 219
Hill, Steady State. 219
Lineweaver-Burk 220
Logistic Growth. 220
Lorentzian Distribution 220
Lorentzian Power 1 221
Lorentzian Power 2 221
Michaelis-Menten 222
Nernst 222
Parabola, Standard. 222
Parabola, Variance-Mean 222
Poisson 223
Polynomial 223
Straight Line, Origin at Zero 223
Straight Line, Standard 224
Voltage-Dependent Relaxation 224
Constants 224

A. References

Primary Sources. 225
Further Reading 228

B. Troubleshooting

Software Problems 231
Hardware Problems 231
Service and Support. 232

C. Resources

Programs and Sources 233

Index. 235

1. Introduction

Welcome to pCLAMP, the data acquisition and analysis software suite from the Axon Instruments Cellular Neurosciences product line of Molecular Devices Corp. Designed for a variety of experiments, pCLAMP 10 is the latest version of a software package that has become the standard for electrophysiological experimentation and analysis. The flexibility that pCLAMP offers allows researchers to adapt it to many uses outside its traditional applications in electrophysiology.

pCLAMP 10 consists of:

- Clampex 10, for data acquisition and production of stimulus waveforms.
- Clampfit 10, for data analysis.
- AxoScope 10, for background chart recording.
- MiniDigi, a two-channel digitizer.

Clampex is a versatile and powerful tool for acquiring digitized data of all types. While excellent for the acquisition of patch-clamp data, it is not limited to measuring voltage- or current-clamp responses. Clampex can be used to measure any physical parameter that can be linearly converted into a voltage. For example, you can monitor and acquire end-plate currents, measure the fluorescence signal from a photomultiplier tube, measure pressure from a strain gauge, or acquire any other combination of analog signals.

Clampfit is a powerful data analysis program with a wide variety of statistics, analyses, transforms and layout tools for electrophysiological data.

Together, AxoScope and the MiniDigi digitizer provide the functionality traditionally performed by a separate chart recorder—for example, for concurrent background recording.

NEW FEATURES

Clampex

See Chapter 4, *"Clampex Features"*, for further description of the following new features, or consult the online Help.

- **Seal & Membrane Tests**
  Continuously runs in between sweeps, or use as a single resizable window.
- **Split-Clock Acquisition**
  Allows multiple sampling rate changes per sweep.

- **Digital Outputs**
  Controls 8 digital outputs per epoch during a sweep.
- **P/N Leak Subtraction**
  Records both raw and subtracted sweeps.
- **Protocol Editor**
  Protocols specified in time units.
- **Digidata 1440A Support**
  Supports up to 4 Analog Output waveforms.
- **AxoClamp 900A Support**
  Software telegraph support for the AxoClamp 900A.

Clampfit

Clampfit has been updated to accommodate the new Clampex features. Otherwise, there are no other new features.

AXOSCOPE

AxoScope is a subset of Clampex. It provides several continuous data acquisition modes, but has no episodic stimulation mode, and hence no capacity to generate stimulus waveforms. Similarly, it has no Membrane Test, and lacks other advanced features found in Clampex, such as the LTP Assistant, the Junction Potential Calculator and instrument telegraphs. With the MiniDigi digitizer—part of a pCLAMP 10 system—AxoScope can be used as a background chart recorder, running alongside Clampex during experiments.

MINIDIGI 1

The MiniDigi 1 digitizer is a low-noise, two-channel digitizer, designed to function with AxoScope as a simple digital chart recorder. It has two independent, 16-bit analog inputs, each of which provides digitization at up to 1 kHz. The MiniDigi 1 digitizer communicates with the host computer (and is powered) through a USB 1 interface.

The MiniDigi 1A digitizer was originally bundled with pCLAMP 9. For users upgrading from previous versions of pCLAMP, or purchasing new copies of pCLAMP 10, the MinDigi digitizer is included at no extra charge.

Filtering

The MiniDigi digitizer uses either minmax or analog-like filtering, at your choice. If you select minmax, both the minimum and maximum values in each sample period are sent to the computer.

The analog-style filter is a lowpass antialiasing filter with a cutoff frequency one fifth of the sampling rate.

Interface Description

The MiniDigi digitizer's front panel has two BNC connectors for analog input channels 0 and 1 respectively. The back panel contains a USB connector and an LED to indicate power-on status. The LED slowly blinks to indicate communication with the software driver.

Specifications

Table 1.1: MiniDigi Specifications (analog input).

Number of input channels	2 single-ended
Resolution	16-bit (1 in 65536)
Acquisition rate (per channel)	1 kHz
Input range	-10.000 V to +10.000 V
Maximum allowable input range	-50 V to +50 V
Input resistance	1 MΩ
Gain value	1
Antialias filter (per channel)	Three-pole, 1.5 kHz Bessel

USB Interface

Low-power (< 100 mA), Universal Serial Bus (USB) 1 device. Software driver is compatible with Windows 2000 and XP Pro operating systems.

PCLAMP DOCUMENTATION

- Extensive documentation is provided with pCLAMP, both to help you learn how to use the programs most effectively, and as a reference source for algorithms and other information:
- A thorough step-by-step PDF tutorial guides you through initial hardware/software setup for data acquisition.
- This manual includes general introductory chapters for both Clampex and Clampfit (all AxoScope functionality is included in Clampex) and tutorials to help you get started with these programs. It also includes sections of general discussion on the use of pCLAMP, and algorithms and other reference material.
- Online Help provides specific help on each program command, as well as overview and "How to …" topics.

Setup Tutorial

The pCLAMP installation includes an easy-to-follow guide for the initial setup and configuration of Clampex, "Setting Up Clampex for Data Acquisition". Open the guide from the Clampex 10.0 Tutorial icon placed on the computer desktop during installation.

The guide opens as an Adobe Acrobat PDF file, allowing you to skip to sections of interest if you do not want to go through each step of the tutorial.

The guide has setup instructions for Axon Instruments Axopatch 200B and MultiClamp 700B amplifiers, using a Digidata 1320-series digitizer, but will be of use for any amplifier and digitizer. It takes you through:

- Cabling and telegraph setup.
- Channel and signal configuration—including setting scale factors.
- Basic protocol definition.

It therefore covers everything necessary to start making real recordings.

Online Help

Clampex, Clampfit and AxoScope all have extensive online Help. This provides detailed command-specific help as well as other more general help topics, and some "How to …" topics. Online Help can be accessed in a number of ways:

- Place the cursor over any command in the drop-down menus and press <F1> to open Help at the topic for that command.
- Push the Help button in any dialog box for help on the dialog.
- All toolbuttons have associated tooltips—popup descriptions of the button's function that come up when the cursor is held over the button.
- Open Help > Clampex Help and find topics using the Contents, Index or Search tabs, or use internal topic links.

The heart of the online Help is the "Contents" tab Menu Reference section, which matches the layout of the main window menus. All commands available in the pCLAMP programs appear in the main menus—though many of these commands are accessible via toolbuttons and popup menus as well—and so the Menu Reference section contains an exhaustive list of the Help topics available for each command, dialog box, and dialog box tab.

Online Help also contains an "Exploring Clampex/Clampfit/AxoScope" section, which introduces the main window and each of the specialty windows in the program, with extensive links to related topics within the Help.

In addition, a General Reference section has topics dealing with broader matters relevant to operating the program, but which are not suited to the Menu Reference section.

All online Help topics appear in the Table of Contents, but can also be accessed via the Index or Search.

User Guide

Given that the online Help contains extensive coverage on the use of features, the User Guide focuses on:

- Installation and setup of pCLAMP.
- Definitions and conventions used, and general discussion of data acquisition and analysis.
- Introduction to the main functions of each of the programs.
- Tutorials for data acquisition and analysis.
- Listing and discussion of the analytical tools used by pCLAMP, including algorithms.

OVERVIEW OF USER GUIDE

The chapters are organized as follows:

- **Chapter 1, Introduction** lists features new to pCLAMP 10, and includes a section on the history of pCLAMP.
- **Chapter 2, General Information** includes definitions and conventions, and general discussions of data acquisition and file types.
- **Chapter 3, Installation and Setup** lists the hardware and software requirements and recommendations for pCLAMP. It also includes detailed software installation and setup instructions.
- **Chapter 4, Clampex Features** provides an introduction to the main Clampex features, with special attention to functions new to this version.
- **Chapter 5, Clampex Experiments** includes tutorials designed to guide new users through experimental setup, introducing them to a range of functions available in Clampex. It includes a discussion of the variety of experiments that can be conducted.
- **Chapter 6, Clampfit Features** provides a general introduction to the main Clampfit features, with special attention to functions new to this version.
- **Chapter 7, Clampfit Tutorials** takes the reader step-by-step through analytical procedures common in the evaluation of physiological data.
- **Chapter 8, Clampfit Digital Filters** describes each digital filter and its characteristics.
- **Chapter 9, Digital Spectral Analysis** is an introduction to Fourier analysis and its application to the power spectra of electrophysiological data.
- **Chapter 10, Additional pCLAMP Analyses** details the formulas used and discusses the various analyses in pCLAMP.
- **Chapter 11, Clampfit Curve Fitting** introduces the varieties of fitting methods available, and guides the reader in the choice of appropriate methods.
- **Chapter 12, Clampfit Fitting Functions** describes each of the predefined fitting functions and explains their use, restrictions, and any required data preprocessing.

UTILITY PROGRAMS

Along with the three main programs of the suite (for which icons are placed on the computer desktop in installation), pCLAMP includes a number of utility programs. These are all loaded in ..\Program Files\Molecular Devices\pCLAMP 10.0\. You can open the pCLAMP 10.0 folder in Windows Explorer and double-click on these programs to run them. The utility programs are:

- **Reset to Program Defaults** (clearregistry.exe): This program is more conveniently run from the desktop: Start > Programs > Axon Laboratory > pCLAMP 10.0 > Reset to Program Defaults. Use it when you encounter strange behavior in the program, or just want to set various Windows registry components back to their factory default settings.
- **ABFInfo** (abfinfo.exe): This is a data file property viewer. Select ABF data files and view their header information in terse, normal or verbose modes. Several file headers can be viewed at once for comparison, but the header information cannot be edited. You can also view file headers one at a time in Clampex and Clampfit, by opening an ABF file and then selecting File > Properties.
- **DongleFind** (DongleFind.exe): A simple application that checks the computer for Axon Instruments software security keys (dongles), including network keys. The application can be set to report a range of information about the keys it finds.

HISTORY OF PCLAMP

The first pCLAMP applications for controlling and analyzing electrophysiological experiments with computers emerged at the California Institute of Technology (Caltech) in 1973. They were originally used for kinetic studies on nicotinic acetylcholine receptors and on voltage-sensitive currents. These early versions were converted to PC-compatible software in mid-1982 and have been in use since 1983 (Kegel *et al.*, 1985).

In 1984, with a view toward serving the entire community of cellular neurobiologists and membrane biophysicists, California Institute of Technology (Caltech) licensed the package to Axon Instruments, Inc. for continued development. pCLAMP continued to evolve into an extremely powerful suite of applications that were used by a broad spectrum of researchers in electrophysiology.

In 1998, the first Windows version of pCLAMP 7 was released. Full conversion from MS-DOS to Windows was completed in 2002 and included support for a variety of synaptic data (long-term potentiation/depression [LTP/LTD] and minis), as well as action potentials.

In 2004, Molecular Devices Corp acquired Axon Instruments, Inc., and released pCLAMP 10 in 2006, with enhanced data acquisition features.

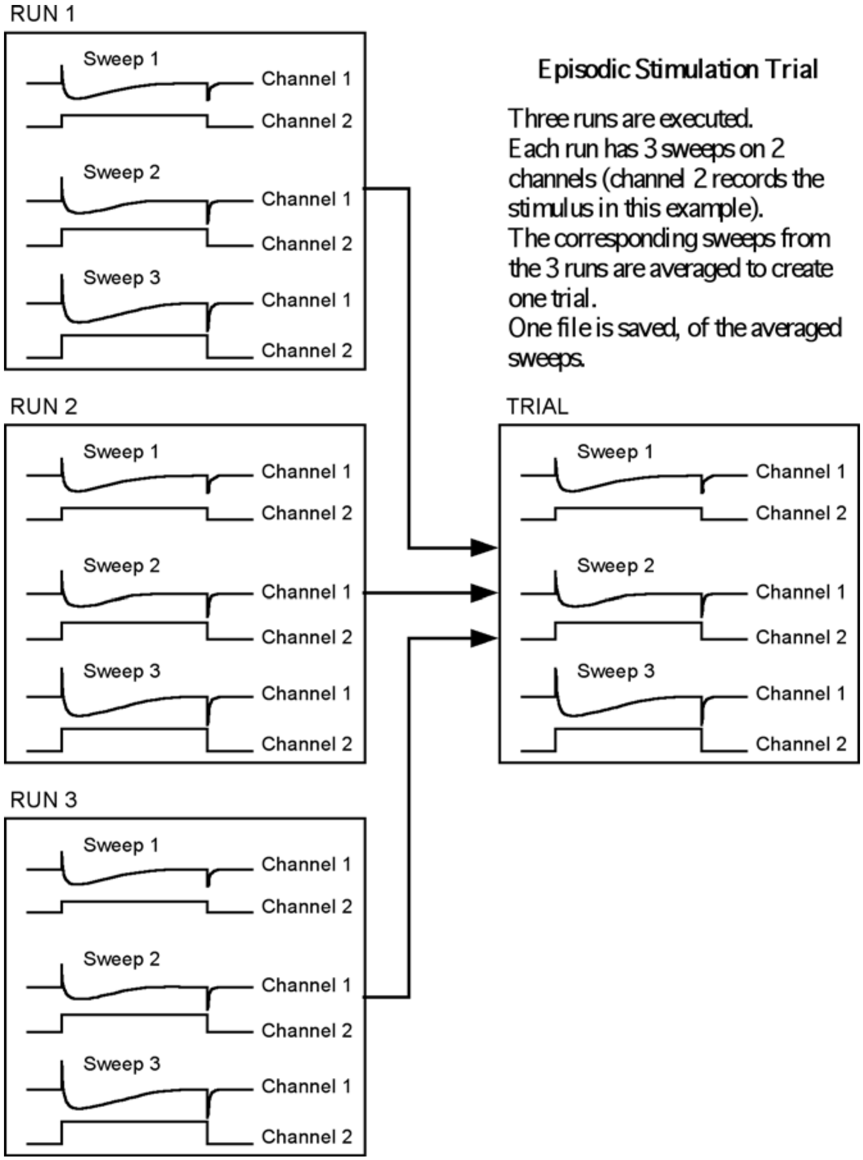
2. Description

Starting with definitions of many of the terms used in pCLAMP, this chapter contains information of a general nature likely to be useful for pCLAMP users. Middle sections in the chapter discuss electrophysiology terminology conventions, and there is a discussion of the theory behind sampling. The final two sections list pCLAMP file types and a range of pCLAMP “vital statistics”.

DEFINITIONS

A number of terms in standard use in Axon Instruments/Molecular Devices applications are defined below:

- A **waveform** consists of a series of analog voltage steps, ramps and/or trains of pulses, or arbitrary data in a file, generated on an output signal in order to stimulate a cell. Also termed the “command waveform” or “stimulus waveform”, it can also have digital outputs associated with it.
- An **epoch** is a subsection of a waveform that can be defined as a step, ramp, or pulse train, and increased or decreased incrementally in amplitude and/or duration from sweep to sweep within a run.
- A **sample** is the datum produced by one A/D (analog-to-digital) conversion or one D/A (digital-to-analog) conversion. In analysis contexts, samples may be referred to as points (see below).
- A **sweep** is the digitized data from all input signals for a defined number of samples. A sweep can contain up to one million samples, with all signals multiplexed at equal time intervals. A command waveform can be concurrently output during a sweep. Sweeps were known as episodes in older versions of pCLAMP. See Figure 2.1 for illustration of the relationship between runs, sweeps, channels and trials.
- A **run** is a set of sweeps. Sweeps within a run may all be the same, or they can be configured to have amplitude and/or duration changes from sweep to sweep. A run can contain up to 10,000 sweeps. If multiple runs are specified, all sets of sweeps are averaged together to produce a single averaged set of sweeps. See Figure 2.1 for illustration of the relationship between runs, sweeps, channels and trials.
- A **trial** is the data digitized from one or more runs, and saved as a single file. SeeFigure 2.1 for illustration of the relationship between runs, sweeps, channels and trials.



Episodic Stimulation Trial

Three runs are executed.
Each run has 3 sweeps on 2 channels (channel 2 records the stimulus in this example).
The corresponding sweeps from the 3 runs are averaged to create one trial.
One file is saved, of the averaged sweeps.

Figure 2.1: Clampex data structure showing relationship between runs and trial.

- A **trace** is a continuous set of data samples from a single input signal. When data are displayed as sweeps, each trace represents a sweep within that signal.
- A **point** is a single datum in a data file, similar to sample, above, although points can be created in a file without having been converted from an analog signal by an A/D converter.

- A **channel** is a physical connection through which analog and digital signals are received or transmitted. Channels are identified in pCLAMP by the name of the digitizer port where connection is made: e.g. Analog IN #0, Digital OUT #5.
- A **signal** is a set of name, unit, scale factor and offset, according to which:

  a Voltage inputs received at the analog input ports of the digitizer are represented in Clampex as the physical parameter (unit) actually being read by the amplifier or transducer, with correct scaling and offset, and a user-selected name.

  b Voltage outputs generated through the digitizer's analog output ports are represented in Clampex as the physical parameter (unit) actually being delivered to the preparation by the amplifier or transducer, with correct scaling and offset, and a user-selected name.

  In Clampex, numerous signals can be configured in association with each analog input and output channel (in the Lab Bench). A specific signal is assigned to a channel in the protocol configuration.
- A **protocol** is a set of configuration settings for a trial. It defines the acquisition mode, the trial's hierarchy (i.e. the number of sweeps per run, and runs per trial), the sampling rate, the definition of the waveform, and many other options as well, which can all be saved into a *.pro protocol file.
- An **experiment** can be composed of several different protocols, and thus may result in several data files. In the context of sequencing keys where protocols can be assigned to keys and also linked to follow one another the *.sks files which define the keys and linkages can be said to define an experiment. Configurations created in the LTP Assistant, which also result in *.sks files, are similarly called experiments.
- An **event** is a discrete response of biological activity, usually relatively short, within an input signal. It can be characterized by event detection, and extracted for further data analysis.
- The **baseline** in an episodic sweep consists of the initial and ending points of the trace, during which the holding level is output. Or, it is the level in an input signal that a trace maintains during periods that no events occur.
- A **peak** is a point in a trace of local maximum deviation from the baseline. Peaks can be positive (above the baseline) or negative (below the baseline).
- The **rise** is that part of the trace that, in an event, goes from the direction of the baseline to the peak. In the case of a negative peak, the rise is a downwards movement of the trace. In previous versions of pCLAMP, rising phases were sometimes referred to with the term left, as in "greatest left slope".
- The **decay** is that part of the trace that, in an event, goes from the direction of the peak back to the baseline. In the case of a negative peak, the decay is an upwards movement of the trace. In previous versions of pCLAMP decay phases were sometimes referred to with the term right, as in "greatest right slope".

- The **mode** most commonly referred to in pCLAMP is the data acquisition mode, set in the Protocol Editor. This determines how data acquisition is triggered and stopped, and whether it is accompanied by a command waveform (see Data Acquisition Modes below).

  Mode can also refer to amplifier mode, which is the amplifier state of current clamp, voltage clamp, I = 0, etc.

- **Acquisition** is the digitization of data by Clampex. Acquired data can be displayed in the Scope window at the same time they are recorded to disk, or viewed without recording.
- **Electrode resistance** (R_e), also called pipette resistance (R_p), is the resistance due to the electrode. It does not include resistance due to environmental current-impeding factors near the electrode tip, e.g. cellular debris, air bubbles, poorly conducting solution etc.
- **Access resistance** (R_a) is the sum of the electrode resistance and resistance due to current-impeding factors near the electrode tip, e.g. cellular debris, etc. Access resistance is sometimes called series resistance (R_s). This is the term used on Axon-made amplifiers.
- **Membrane resistance** (R_m) is the resistance across the cell membrane.
- **Total resistance** (R_t) is the sum of membrane resistance and access resistance. When an electrode seals against the membrane, if the seal is successful, i.e. a gigohm seal, access resistance is a negligible component of the total resistance, so the total resistance is effectively equal to the seal resistance.
- **Seal resistance** is the resistance afforded by the seal between the electrode tip and the cell membrane.

DATA ACQUISITION MODES

Clampex provides five distinct data acquisition modes:

Gap-free Mode

This mode is similar to a chart or tape recorder, where large amounts of data are passively and continuously digitized, displayed, and saved without any interruptions to the data record.

Variable-length Events Mode

Data are acquired for as long as an input signal has passed the threshold level, or for as long as an external trigger is held high. This mode is ideal for experiments such as recording of single-channel currents that are in a closed state for long periods of time and contain periods of random bursting.

Fixed-length Events Mode

Data are acquired for same-length sweeps whenever an input signal has passed the threshold level, or when an external trigger occurs. This mode is ideal for recording synaptic events, action-potential spikes or other constant-width events. If during one fixed-length event a second trigger occurs, a second fixed-length event is started. The two events have overlapping data until the first event ends. In this way no events are lost, and each event has the same length.

High-speed Oscilloscope Mode
In this mode, data are acquired in sweeps, as with a standard oscilloscope. The input sweep can be triggered by either an external trigger, an autotrigger, or the input signal crossing a threshold level.

High-speed oscilloscope mode resembles fixed-length event mode, except that extra triggers occurring during a sweep do not initiate additional sweeps.

Episodic Stimulation Mode
An analog waveform, holding level, and/or digital pulses are output, while data are simultaneously acquired in fixed-length sweeps. Each sweep is non-overlapping and can be triggered by an internal timer or by a manual or external pulse.

Episodic stimulation mode is useful for studying voltage-activated currents using the whole-cell patch-clamp configuration. For example, Clampex can be used to drive a membrane potential to various potentials in controlled amplitude and duration increments or decrements. The cellular response to these test potentials is simultaneously acquired. Special features include pre-sweep trains, online leak current subtraction, online peak detection and statistics, and an online derived-math channel. Online statistics can be used to chart peak measurement values in real time.

TERMS AND CONVENTIONS IN ELECTROPHYSIOLOGY

Current and Voltage Conventions

In many cases, Clampex is used to record membrane potential and current. There are various conflicting definitions of current and voltage polarities, so we take the opportunity to discuss here the conventions used for Axon Instruments products. This information is also presented in various instrument manuals and other publications provided by Axon Instruments/Molecular Devices.

Positive Current
In this discussion, and in all amplifiers manufactured by Axon Instruments/Molecular Devices, the term positive current means the flow of positive ions out of the headstage into the micropipette and out of the micropipette tip into the preparation.

Inward Current
Inward current is current that flows across the membrane, from the outside surface to the inside surface.

Outward Current
Outward current is current that flows across the membrane, from the inside surface to the outside surface.

Positive Potential

In this discussion, and in all amplifiers manufactured by Axon Instruments/Molecular Devices, the term positive potential means a positive voltage at the headstage input with respect to ground.

Transmembrane Potential

The transmembrane potential (V_m) is the potential at the inside of the cell minus the potential at the outside. This term is applied equally to the whole-cell membrane and to a membrane patch.

Depolarizing/Hyperpolarizing

The resting V_m value of most cells is negative. If a positive current flows into the cell, V_m initially becomes less negative. For example, V_m might shift from an initial resting value of -70 mV to a new value of -20 mV. Since the absolute magnitude of V_m is smaller, the current is said to depolarize the cell (i.e. it reduces the "polarizing" voltage across the membrane). This convention is adhered to even if the current is so large that the absolute magnitude of V_m becomes larger. For example, a current that causes V_m to shift from -70 mV to +90 mV is still said to depolarize the cell. Stated simply, depolarization is a positive shift in V_m. Conversely, hyperpolarization is a negative shift in V_m.

Whole-Cell Voltage and Current Clamp

Depolarizing/Hyperpolarizing Commands

In whole-cell voltage clamp, whether it is performed by TEVC, dSEVC, cSEVC or whole-cell patch clamp, a positive shift in the command voltage causes a positive shift in V_m and is said to be depolarizing. A negative shift in the command voltage causes a negative shift in V_m and is said to be hyperpolarizing.

Transmembrane Potential vs. Command Potential

In whole-cell voltage clamp, the command potential controls the voltage at the tip of the intracellular voltage-recording micropipette. The transmembrane potential is thus equivalent to the command potential.

Inward/Outward Current

In a cell generating an action potential, depolarization is caused by a flow of positive sodium or calcium ions into the cell. That is, depolarization in this case is caused by an inward current.

During intracellular current clamp, a depolarizing current is a positive current out of the micropipette tip into the interior of the cell. This current then passes through the membrane out of the cell into the bathing solution. Thus, in intracellular current clamp, a depolarizing (positive) current is an outward current.

During whole-cell voltage clamp, sodium inward current flows in some cells after a depolarizing voltage step. This current is canceled by an equal and opposite current flowing into the headstage via the micropipette. Thus it is a negative current. When two-electrode voltage clamp was first used in the early 1950s, the investigators chose to call the negative current that they measured a depolarizing current because it corresponded to the depolarizing sodium current. This choice, while based on sound logic, was unfortunate because it means that from the recording instrument's point of view, a negative current is hyperpolarizing in intracellular current-clamp experiments but depolarizing in voltage-clamp experiments.

Because of this confusion, Axon Instruments/Molecular Devices has decided to always use current and voltage conventions based on the instrument's perspective. That is, the current is always unambiguously defined with respect to the direction of flow into or out of the headstage. Some instrument designers have put switches into the instruments to reverse the current and even the command voltage polarities so that the researcher can switch the polarities depending on the type of experiment. This approach has been rejected by Axon Instruments/Molecular Devices because of the real danger that if the researcher forgets to move the switch to the preferred position, the data recorded on the computer could be wrongly interpreted. We believe that the data should be recorded unambiguously.

Patch Clamp

The patch-clamp pipette current is positive if it flows from the headstage through the tip of the micropipette into the patch membrane. Whether it is hyperpolarizing or depolarizing, inward or outward, depends upon whether the cell is "cell attached", "inside out" or "outside out".

Cell-Attached Patch

The membrane patch is attached to the cell. The pipette is connected to the outside surface of the membrane. A positive command voltage causes the transmembrane potential to become more negative, therefore it is hyperpolarizing. For example, if the intracellular potential is -70 mV with respect to 0 mV outside, the potential across the patch is also -70 mV. If the potential inside the pipette is then increased from 0 mV to +20 mV, the transmembrane potential of the patch hyperpolarizes from -70 mV to -90 mV.

From the examples it can be seen that the transmembrane patch potential is inversely proportional to the command potential, and shifted by the resting membrane potential (RMP) of the cell.

A positive pipette current flows through the pipette, across the patch membrane into the cell. Therefore a positive current is inward.

Inside-Out Patch

The membrane patch is detached from the cell. The surface that was originally the inside surface is exposed to the bath solution. Now the potential on the inside surface is 0 mV (bath potential). The pipette is still connected to the outside surface of the membrane. A positive command voltage causes the transmembrane potential to become more negative, therefore it is hyperpolarizing. For example, to approximate resting membrane conditions, say V_m = -70 mV, the potential inside the pipette must be adjusted to +70 mV. If the potential inside the pipette is increased from +70 mV to +90 mV, the transmembrane potential of the patch hyperpolarizes from -70 mV to -90 mV.

From the example it can be seen that the transmembrane patch potential is inversely proportional to the command potential.

A positive pipette current flows through the pipette, across the patch membrane from the outside surface to the inside surface. Therefore a positive current is inward.

Outside-Out Patch

The membrane patch is detached from the cell in such a way that the surface that was originally the outside surface remains exposed to the bath solution. The potential on the outside surface is 0 mV (bath potential). The pipette interior is connected to what was originally the inside surface of the membrane. A positive command voltage causes the transmembrane potential to become less negative, therefore it is depolarizing. For example, to approximate resting membrane conditions, say V_m = -70 mV, the potential inside the pipette must be adjusted to -70 mV. If the potential inside the pipette is then increased from -70 mV to -50 mV, the transmembrane potential of the patch depolarizes from -70 mV to -50 mV.

The membrane potential is directly proportional to the command potential.

A positive pipette current flows through the pipette, across the patch membrane from the inside surface to the outside surface. Therefore a positive current is outward.

Summary

1 Positive current corresponds to:

Cell-attached patch	patch inward current
Inside-out patch	patch inward current
Outside-out patch	patch outward current
Whole-cell voltage clamp	outward membrane current
Whole-cell current clamp	outward membrane current

2 A positive shift in the command potential is:

Cell-attached patch	hyperpolarizing
Inside-out patch	hyperpolarizing
Outside-out patch	depolarizing
Whole-cell voltage clamp	depolarizing

3 The correspondence between the command potential (V_{cmd}) and the transmembrane potential (V_m) is:

Cell-attached patch	V_m = RMP - V_{cmd}
Inside-out patch	V_m = -V_{cmd}
Outside-out patch	V_m = V_{cmd}
Whole-cell voltage clamp	V_m = V_{cmd}

THE SAMPLING THEOREM IN CLAMPFIT

The sampling theorem states that an analog signal can be completely reproduced by regularly spaced samples if the sampling frequency is at least 2 times that of the highest frequency component in the signal. Thus the minimum sampling interval T is given by

$$T = \frac{1}{2f_h}$$

where f_h is the highest frequency component. For example, if the highest frequency component in an analog signal is 5000 Hz, the sampling rate should be at least 10,000 times per second if the signal is to be faithfully reproduced.

The maximum frequency f_h in an analog signal is generally referred to as the Nyquist frequency. The minimum sampling rate of $2f_h$ samples per second that is theoretically required to accurately reproduce the analog signal is referred to as the Nyquist rate. If the sampling rate is less than the Nyquist rate, then two types of errors are introduced. The first is that high frequency information will be irretrievably lost. The second is the introduction of artificial low frequency components, referred to as aliasing.

Aliasing is especially problematic if there are periodic components in the analog signal. This is illustrated in Figure 2.2, which represents a 2500 Hz sine wave sampled at a rate of 2000 Hz (two-fifths the Nyquist rate). The sampling points are shown as dark squares. Note that the reconstructed 500 Hz waveform (heavy line) is only one-fifth the frequency of the original signal.

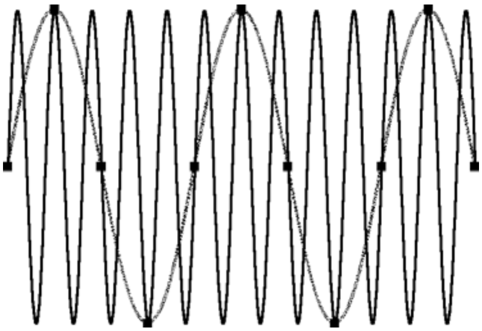


Figure 2.2: Illustration of aliasing.

In the real world, information regarding the exact spectral composition of analog signals is rarely available, especially in the presence of noise. To avoid distortion of the digital signal by frequencies that are above the Nyquist limit, a value of f_h should be selected in accordance with the particular experimental requirements, and the analog signal should be lowpass filtered before sampling to reject frequencies above f_h. Filters applied in this way are known as antialiasing filters or guard filters.

In practice, it is common to sample at a rate significantly faster than the minimum rate specified by the sampling theorem. This is known as oversampling. Exactly how much oversampling should be used depends on the experiment.

OPTIMAL DATA ACQUISITION

Computer-based data acquisition extends the range of the types, complexity and size of experiments that can be readily performed in the laboratory. To use these tools effectively, several key concepts that underpin all computerized data acquisition systems should be understood, and these are discussed briefly in the following sections.

Analog Data

The fundamental property of analog data is that it is continuous. Analog data can be obtained from transducers recording a wide variety of properties, including (but not limited to) voltage, current, pressure, pH, speed, velocity, light, sound levels, etc. The amplitudes of any of these signals may vary over wide ranges, and the increments are infinitesimally small. While analog signals can be recorded directly by transferring them from an analog output to an analog recording device (e.g. chart recorder, FM tape recorder, etc.), analysis and reproduction of these records always involves some signal degradation, due to the effects of noise or distortion.

Analog to Digital Conversion

The loss of analog signal fidelity can be minimized by the effective use of analog-to-digital conversion. This is the process of converting an analog signal into a digital representation. Such a representation can be stored on a computer disk, printed page, etc., without subsequent signal degradation. The popularity of the audio compact disc is based on the effective use of analog-to-digital conversion to store and reproduce the music recorded on it. The digital representation can then be replayed precisely, as often as desired, without the introduction of noise.

While seemingly obvious, the effective use of analog-to-digital conversion requires that one consider several conflicting goals carefully. In simplest terms, one must decide how best to preserve the fidelity of the analog signal, using an affordable quantity of recording media. Since an analog signal is continuous, A/D conversion inherently yields an approximation of the original data. The goal in A/D conversion is to make reasonable assumptions with respect to the necessary temporal and amplitude resolutions that are required to reproduce the original analog signal, and then to choose and set the acquisition parameters appropriately.

Temporal Resolution

Does the digital representation of the original signal faithfully reproduce the response of the analog signal in the time domain? It is obvious that a signal with a 10 Hz component will not be reproduced well by sampling the signal once per second (1 Hz). While an acquisition at 1,000 Hz is intuitively adequate, it uses unnecessary resources for the storage of the acquired signal.

So how fast do you need to sample to adequately reproduce the temporal characteristics of the analog signal? The Nyquist sampling theorem states that if a DC signal is sampled at a rate that is twice the analog bandwidth of the signal, the sampled values can reproduce the original signal. Thus, if the sampling rate is 1/*t* (where *t* is the sampling interval), the signal must have no frequency components greater than 1/(2*t*). Sampling at a frequency of twice the analog bandwidth is the theoretical minimum required to reproduce the source signal.

If appropriate sampling frequencies are not used, two potential errors are introduced. The most obvious one is that the high frequency information is lost; the less obvious error is the introduction of aliasing. Aliasing is the introduction of a spurious low-frequency signal. For those old enough to remember 8 mm home movies (or fortunate enough to have video frame grabbers in their computers), frame rates of 8–12 frames per second often yield cars whose wheels appear to be turning backwards (while the car is moving forward!). This illusion is the effect of aliasing on our visual perception of the wheel motion.

Aliasing, Filtering, and Oversampling

In practice, it is common to sample at a rate significantly faster than the minimum rate specified by the sampling theorem. This is known as oversampling. Exactly how much oversampling should be used depends upon the type of experiment.

For experiments where the data will be analyzed in the frequency domain (e.g. noise analysis, impedance analysis), it is common to oversample only modestly. The main concern is to prevent aliasing. An antialiasing filter is introduced between the signal source and the analog-to-digital converter to control the bandwidth of the data.

The factor of twice the analog bandwidth required by the sampling theorem is only applicable if the antialiasing filter is ideal, i.e. the gain in the pass-band is unity and in the stop-band it abruptly changes to zero. Ideal filters cannot be realized, although they can be closely approximated. For frequency-domain analysis, it is common to use sharp cutoff filters such as Butterworth or Chebyshev realizations. Sampling is typically performed at 2.5 times the filter bandwidth. For example, if the data are filtered at 10 kHz, they should be sampled at about 25 kHz. Slower sampling rates are unacceptable. Faster sampling rates are acceptable, but offer little advantage, and increase the storage and analysis requirements.

For experiments where the data will be analyzed in the time domain (e.g. pulse analysis, I-V curves), greater oversampling is required. This is because reconstruction of the analog signal requires not only an ideal antialias filter, but also an ideal reconstruction filter. The simplest and most common reconstruction filter is to join each sample by a straight line. Other techniques can be used, such as cubic-spline interpolation, but because of their much heavier computational requirements they are infrequently used.

There is no golden rule to determine how fast to sample data for time-domain analysis, but in general, 5 times the analog bandwidth is common, and 10 times is regarded as good.

Amplitude Resolution

The amplitude resolution of an A/D converter corresponds to the smallest increment in signal that it can resolve. This resolution is a result of two properties of the converter hardware: the number of “bits” in the conversion, and the full-range voltage input that the A/D converter can handle. Most high-speed A/D converters (e.g. those supported by pCLAMP) are binary devices with 12 to 16 bits of resolution. The number of possible A/D values is a power of two, often referred to as the number of bits. Commonly, these values are:

8-bit converter	= 28 = 256 values
12-bit converter	= 212 = 4,096 values
16-bit converter	= 216 = 65,536 values

The full voltage range of most A/D converters is typically ±10 V. The amplitude resolution is defined by the full-scale input range divided by the number of sample values (or quanta). Thus, for a 12-bit system, the amplitude resolution is 20 V/4,096 quanta or 4.88 mV/quanta. For a 16-bit system, the resolution is 0.305 mV/quanta.

If one wants to obtain the best resolution of the source signal, the need for amplifiers and/or preamplifiers to scale the input signal appropriately becomes apparent. The goal is to have the input signal use as much as possible of the input voltage range of the converter, so that the resolution of the data signal can be as precise as possible. Thus, for a biological signal that varies over the range of ±100 mV, amplification with a gain of up to 100 is needed to fill the ±10 V data acquisition range.

FILE FORMATS

Binary Data

Clampex acquires and stores data in the Axon Binary Format file format (ABF). Binary encoding is compact, so data files do not occupy more space on disk than is necessary.

There are two types of ABF files:

- **Binary Integer:** When raw data are digitized, acquired and stored to a data file, they are saved as binary integer numbers.
- **Binary Floating Point:** When a data file is opened into an Analysis window for review, the data are internally converted to floating point numbers. This increases the amount of precision, which is necessary for applying certain mathematical operations to the data. When saving a data file from an Analysis window, you can save it in either integer format or floating-point format.

Clampex and Clampfit can read integer or floating point ABF files. All pCLAMP 10 programs read all previous versions of pCLAMP ABF data files. In addition, there are several third-party software packages that directly support our integer binary data file format (see the Molecular Devices web site for details).

Text Data

Clampex can also save data files in the Axon Text File file format (ATF), which is an ASCII text format. Thus, ATF files are easily imported into spreadsheet, scientific analysis, and graphics programs, and can also be edited by word processor and text editor programs.

Be aware that data stored in a text format occupies much more disk space than data stored in a binary format, and does not include the full header information in an ABF file, so the ATF format is only recommended for transferring data into other programs that do not support the ABF file format.

Plain Data Files

Clampex can read and write binary and text files that do not have a header. When such files are read, the header information must be manually re-created by the user. For this reason, plain binary and plain text files are not recommended, but they can sometimes be useful for data interchange between third-party programs and Clampex.

Header Information (ABFInfo)

The ABFInfo utility allows you to inspect the header information of ABF binary data files and ABF protocol files. The header information includes the entire protocol used to acquire the data, as well as information specific to the file, such as the time of acquisition, and the actual number of samples acquired.

Programming Information

The file format specifications are described in detail in ABFInfo and in the ABF File Support Pack (FSP) for programmers. This is available from the Molecular Devices web site. The Axon FSP includes library routines written in C^{++}, and supports the reading and writing of ABF- and ATF-format files.

Axon Instruments' long-standing role in electrophysiology data acquisition systems has led to the support of several third-party programs for the analysis of pCLAMP data. Individual investigators can sometimes find such programs suited for their particular needs. In order for Clampex binary integer data files to be recognized by software packages compatible with pCLAMP 6 binary data files, it may be necessary to record the data with the Clampex Program Option configured to "Ensure data files and protocols are compatible with pCLAMP 6", and to change the data file's extension from "abf" to "dat".

With pCLAMP 10, the ABF file format has changed, so that third-party programs compatible with pCLAMP 9 data files might not be able to read pCLAMP 10 data files. In this case, pCLAMP 10 data files will need to be exported as pCLAMP 9 compatible files.

Other File Formats

pCLAMP uses a number of specialized file formats for files generated in specific contexts. These are:

- **Axon Layout File (ALF):** For data saved from Clampfit's Layout window, where data can be arranged for presentation. The ALF file format has changed in Clampfit 10 and is not compatible with earlier versions.
- **Data File Index (DFI):** For files saved from the Data File Index window in both Clampex and Clampfit. The Data File Index window is a file management tool.
- **Junction Potential Calculation (JPC):** For files saved from Clampex's Junction Potential Calculator.
- **Rich Text Format (RTF):** This is a general-use text format not specific to pCLAMP. Lab Book files are saved in RTF.
- **Protocol File (PRO):** For protocol settings in the Clampex Protocol Editor. These files are stored in ABF format, like the binary data files.
- **Results File (RLT):** For files saved from the Results window in Clampex or Clampfit. Any graphs generated from Results window data are also saved in the RLT file.
- **Sequencing Key File (SKS):** For sequencing key sets saved from Configure > Sequencing Keys in Clampex. Sequencing keys allow you to link protocols and the Membrane Test to follow in predefined sequences.
- **Search Protocol File (SPF):** For search configurations saved from the three Clampfit event detection searches (single-channel, threshold and template).
- **Statistics File (STA):** For data from the Clampex Online Statistics window. If opened in Clampfit, these files open both in a Statistics window and as text in the Results window.

PCLAMP QUANTITATIVE LIMITS

Maximum and minimum settings for a range of parameters in Clampex and Clampfit are presented in the following tables:

Clampex

Table 2.1: Clampex settings.

Location	Parameter	Quantity
Protocol Editor Mode > Rate tab	Max. runs per trial	10,000
	Max sweeps per run	10,000
	Max. samples per sweep	1,032,258
	Max. samples per trial	2,147,483,647 including unacquired samples between sweeps
	Max. data file size	4 GB
	Max. sampling rate (kHz)	500 (DD1322A, DD1321A) 250 (DD1440, DD1320A)
	Min. sampling rate (Hz)	1
	Min. sampling interval (μs)	2 μs (DD1322A, DD1321A) 4 μs (DD1440A, DD1320A)
	Min. start-to-start interval	Sweep length (i.e. zero delay between end of one sweep and start of next)
	First and last holding periods	1/64 of sweep length each
Protocol Editor Inputs tab	Max. analog input channels	16 (15 if a Math signal or P/N leak subtraction is enabled)
Protocol Editor Outputs tab	No. analog output channels	4 (DD1440A) 2 (DD1320 series)
Protocol Editor Statistics tab	Max. samples in boxcar filter smoothing window	21 (i.e. 10 on either side of each sample)
Protocol Editor Wave tab	Max. epochs	10
	Max. waveform channels	4 (DD1440A) 2 (DD1320 series)

Location	Parameter	Quantity
Protocol Editor Stim tab	Max. characters in User List	512 (including commas; unlimited with Repeat)
	Max. Leak Subtraction subsweeps	8
	Max. pre-sweep train pulses	10,000
Scope window	Max. open windows	4
Analysis window	Max. open windows	16
Sequencing keys	Max. defined keys	50, plus additional 32 MultiClamp amplifier-mode keys

Clampfit

Table 2.2: Clampfit settings.

Location	Parameter	Quantity
Analysis window	Max. open windows	No limit
	Max. signals	12
	Max. characters in Select Sweeps user-entered list	512 (including commas)
	Max. samples transferred to Results or Graph windows	1,000,000
Fitting (Analysis, Graph, & Results windows)	Max. points that can be fitted	1,000,000
	Max. function terms	6
	Max. power	6
	Max. custom function parameters	24
	Max. independent variables in a custom function	6
	Max. dependent variables in a custom function	1
	Max. points for functions containing a factorial term	170

Location	Parameter	Quantity
Results window	Max. rows	1,000,000
	Max. imported columns	8,000
	Number of sheets	20
	No. of operations that can be undone	10
	Max. rows for Create Data	110,000
Event detection	Max. search categories (peak-time events)	9
	Max. levels (single-channel search)	8
Graph window	Max. plots	1000

3. Setup

COMPUTER SYSTEM

Table 3.1: Computer system requirements.

Minimum System Requirements	Recommended System Requirements
Windows-compatible PC with 1 GHz Pentium class CPU[a]	Windows-compatible PC with 2 GHz (or higher) Pentium class CPU[a]
Windows 2000 operating system	Windows XP Pro SP2 operating system
128 MB RAM	256 MB RAM or higher
CD-ROM drive (for installation)	CD-ROM drive (for installation)
1024 X 768 display system for Clampfit 800 X 600 display system for Clampex	1024 X 768 display or higher
USB 1.0 port	USB 2.0 port for Digidata 1440 digitizer USB (1.0 or 2.0) port for security key
PCI slot (full height)[b]	PCI slot (for Digidata 1320 series digitizer)[b]

a.Multiple processor systems are not supported.
b.Note that “slimline” computer cases are not compatible with standard-size PCI cards

Software Protection Key

A software protection “key”, commonly known as a “dongle”, is provided with Clampex. The USB key is a small device (2″ x 0.5″ x 0.25″) that plugs into the computer’s USB port.

The key is required to configure Clampex to control the digitizer. If the key is not installed, Clampex runs in Demo mode only, restricting you to simulated data.

Signal Connections

Clampex uses the following BNC connections on the Digidata 1440 series data acquisition systems:

Table 3.2: Clampex Digidata 1440 BNC connections.

Clampex Signals	Digidata 1440 BNC Sections	Digidata 1440 BNC Names
4 Analog OUT Channels	ANALOG OUTPUTS (Front)	0–3
16 Analog IN Channels	ANALOG INPUTS (Front)	0–15
Telegraphs (Gain, Frequency, C_m Capacitance)	TELEGRAPH INPUTS (Rear)	0–3
8 Digital Outputs	DIGITAL OUTPUTS (Front)	0–7
1 Digitizer START Input	(Front)	START
1 External Tag Input	(Front)	TAG
Scope Trigger Output	(Front)	SCOPE

Clampex uses the following BNC connections on the Digidata 1320 series data acquisition systems:

Table 3.3: Clampex Digidata 1320BNC connections.

Clampex Signals	Digidata 132x BNC Sections	Digidata 132x BNC Names
4 Analog OUT Channels	ANALOG OUT (Front)	0–1
16 Analog IN Channels	ANALOG IN (Front)	0–15
Telegraphs (Gain, Frequency, C_m Capacitance)	TELEGRAPH INPUTS (Rear)	0–4
8 Digital Outputs	DIGITAL OUT (Front)	0–3
	DIGITAL OUTPUTS (Rear)	4–7
1 Digitizer START Input	TRIGGER IN (Front)	START
1 External Tag Input	TRIGGER IN (Front)	TAG
Digidata 1322A Scope Trigger Output	(Rear)	TRIGGER OUTPUT
Digidata 1320A/1321A Scope Trigger Output	(Rear)	ADC CLOCK OUTPUT

Analog Output Signals

Clampex uses an analog output channel from the digitizer to control the holding level and/or command waveform of an experiment. For example, the ANALOG OUT #0 channel would be connected via a BNC cable to a microelectrode current/voltage clamp amplifier's External Command Input. If the amplifier has an internal command generator, be sure to switch to external command control. The other analog output channels can be used to control a separate holding level or output other command waveforms.

Analog Input Signals

The output signals from an amplifier connect to the digitizer's analog input channels. With Clampex, these analog signals are digitized, displayed on the computer screen, and optionally saved to a data file on the hard disk.

Digital Outputs

Clampex supports eight TTL-compatible digital outputs. All of these can be configured with bit patterns that coincide with the command waveform, enabling you control other instruments, such as a solution changer or a picospritzer. All eight can be configured with a holding pattern, or changed via a sequencing key. Clampex can also output a dedicated scope trigger to synchronize signal digitization with an oscilloscope.

Digital Inputs

The Clampex trigger inputs allow other instruments to externally trigger the start of acquisition, as well as to trigger the insertion of time, comment or voice tag information directly into the data file. A TTL-compatible digital input is required.

Telegraphs

Clampex can be configured to receive "telegraphs" from many amplifiers, reporting such amplifier settings as the variable gain, lowpass filter, and whole-cell capacitance compensation.

Older model amplifiers have a BNC port for each type of telegraph they generate. These must be cable-connected to the digitizer. Digidata 1320 and 1440 series digitizers have dedicated telegraph BNC ports. MultiClamp 700 and AxoClamp 900A amplifiers are computer-controlled, so telegraphs are passed directly to Clampex digitally, with no additional cabling required. See *"Telegraphs"* on page 27 for more about telegraphs in Clampex.

SOFTWARE SETUP AND INSTALLATION

Prior to loading the pCLAMP CD, you should exit all other Windows programs.

Windows 2000/XP Pro

Clampex is a full 32-bit program that runs under Windows 2000 and XP Pro. The pCLAMP Setup program automatically detects which operating system is running, and loads the correct files.

Automatic CD Loading

When the pCLAMP CD is inserted into the CD-ROM drive, it automatically loads and displays the Setup program. This process may take several seconds to complete.

If you prefer to manually start the pCLAMP Setup program, use the Windows Explorer to go to the CD-ROM drive, and then double-click on the Setup icon.

Installing pCLAMP—Standard Installation

The following procedure explains how to install the pCLAMP suite.

→ Run the Setup pCLAMP program from the CD ROM.

→ In the Choose Destination dialog box, you can change the destination drive and directory where pCLAMP is installed.

→ The amount of hard disk space required for this installation and the amount of space available on the hard disk are displayed. The default Axon Laboratory Program Folder is created and program icons are added to it. You can rename the Program Folder or select one of the existing folders. Setup will then copy the pCLAMP files to the computer.

→ For Clampex to run properly, you need to restart the computer.

→ Before rebooting, remove the pCLAMP CD from the drive.

Uninstalling pCLAMP

To uninstall pCLAMP:

→ Go to Windows Start > All Programs >Axon Laboratory > pCLAMP 10.0

→ Select Uninstall pCLAMP 10.0.

File Locations

User-related files, such as data and parameter files, are stored in their own folders in ..\Documents and Settings\[user name]\My Documents\pCLAMP\…

System-related files, such as for the Lab Bench, System Lab Book, and user-defined telegraphs, are stored in the folder ..\Documents and Settings\All Users\Application Data\Molecular Devices\pCLAMP\.

Program application files are stored by default in the folder ..\Program Files\Molecular Devices\pCLAMP 10.0.

DIGITIZER CONFIGURATION IN CLAMPEX

Once you have loaded pCLAMP and connected the digitizer to the computer, you must configure Clampex to communicate with the digitizer. This is done from the Configure > Digitizer dialog.

Detailed instructions on digitizer configuration are included in the tutorial “Setting Up Clampex for Data Acquisition”, which was loaded onto the computer in the pCLAMP installation. The tutorial can be opened from the icon on the desktop. Instructions here summarize those provided in the tutorial.

Demo Mode

When Clampex is first installed, it is in “Demo” mode, allowing you to experiment with the program without being connected to a digitizer. The demo digitizer is like having a built-in signal generator. It creates signals derived from episodic protocols and adds noise, making it perfect for creating sample data files. Or, from the Configure > Digitizer dialog with “Demo” selected, click on the Configure button to alter the demo data output in non-episodic acquisition modes.

Configuring the Digidata 1320 or 1440 Series Digitizers in Clampex

Connect the Digidata digitizer to the computer. If you have a Digidata 1440A digitizer, the Windows Found New Hardware Wizard is displayed. Work through the prompts until Windows has installed the digitizer. No separate driver disk is needed, so it is recommended that you automatically search the hard disk for the driver.

- Start Clampex by clicking Start > All Programs > Axon Laboratory > pCLAMP 10.0 > Clampex 10.0.
- Open the Configure > Digitizer dialog box and select the Change button.
- In the Change Digitizer dialog box select Digidata 1440 Series from the Digitizer Type list.
- Press the Scan button to detect the digitizer. “Available” is displayed and the OK button is enabled.
- Click OK to exit this dialog, and click OK again to exit the Digitizer dialog.
- The Digidata digitizer is now ready to perform experiments.

If you receive an error message, refer to the Troubleshooting chapter of the Digidata user manual.

MINIDIGI INSTALLATION

The MiniDigi digitizer can be run with AxoScope, but not Clampex.

- Run the pCLAMP installer before you connect the MiniDigi digitizer to the computer.
- Once pCLAMP has been installed, connect the USB cable to the USB port on the computer and to the MiniDigi digitizer. The Windows Found New Hardware Wizard is displayed. Work through the prompts until Windows has installed the digitizer.
- Start AxoScope by clicking Start > All Programs > Axon Laboratory > pCLAMP 10.0 > AxoScope 10.0.
- Open the Configure > Digitizer dialog box and select the Change button.
- In the Change Digitizer dialog box select MiniDigi from the Digitizer Type list.

- Press the Scan button to detect the digitizer. “Available” is displayed and the OK button is enabled.
- Click OK to exit this dialog box.
- Press the Configure button to open the Configure MiniDigi dialog.
- Select the style of filtering to use.
- “Analog filtering” applies a lowpass antialiasing filter with a cutoff frequency one fifth of the sampling rate.
- “MinMax filtering” takes the minimum and maximum samples in every n samples, where n is determined by the sampling rate.
- To calibrate the MiniDigi digitizer, attach a grounding plug to the Channel 0 BNC, then press the Start button.
- Repeat for Channel 1.
- Click OK to exit this dialog box.
- The MiniDigi digitizer is now ready to perform experiments.

RESETTING PROGRAM DEFAULTS

The Start > All Programs > Axon Laboratory > pCLAMP 10.0 folder contains the utility Reset to Program Defaults, which resets pCLAMP settings back to their default values. This is useful when you feel that you have diverged from the normal setup to a point beyond your control, and you would like to return to the factory defaults. Note that settings for other programs may be displayed in the list of registry items—select the item(s) relevant to your situation.

PRINTING

pCLAMP supports all printers and plotters that have been installed through the Microsoft Windows operating system.

4. Clampex Features

This chapter introduces the major features of Clampex, including an extended discussion of the LTP Assistant. Having read the chapter, the new user can reinforce and extend their understanding by working through the tutorials in Chapter 5, *"Clampex Tutorials"*. More detailed information on the features discussed in this chapter is contained in the Clampex online Help.

CLAMPEX WINDOWS

Clampex has a standard Windows format, with title bar, menus, and toolbars at the top, and a status bar along the bottom. As is typical of many Windows applications, you can choose which toolbars to display (from the View menu) and select the toolbuttons to appear in the toolbars (Configure > Toolbars). There is one dockable component—the Real Time Controls. By default, this opens as a panel attached to the left-hand side of the main window, but can be dragged away to be repositioned like a standard dialog box, or attached to the right-hand side of the main window.

All commands available in Clampex are included in the main menus, although many of them have toolbuttons or can be accessed from right-click popup menus as well. Note that the main menu contents differ according to which of the windows (below) is highlighted.

Besides the main window, Clampex has seven window types:

- Analysis
- Data File Index
- Lab Book
- Membrane Test
- Results
- Scope
- Statistics

Within the main Clampex window, these windows can be maximized, minimized, resized and tiled. Right-clicking in each brings up a popup menu with options specific to the window, including in each case one that opens a Properties dialog box. This allows users to configure the general appearance of the window, and these settings can then be saved as defaults (View > Window Defaults).

The basic roles of each of the window types are as follows (see the online Help system for greater detail):

Analysis Window

The Analysis window displays data that have been saved in a file, for review and measurement. The data are displayed graphically, as traces, with subwindows for each signal stored within the file. Open a file in an Analysis window from File > Open Data.

Data can be displayed as Sweeps, or in Continuous or Concatenated modes, controlled from View > Data Display. In sweep mode you can choose to view any subset of the sweeps (View > Select Sweeps), and toggle between viewing all the sweeps at once, a selected subset, or just one at a time (View > Toggle Sweep List). When more than one sweep is visible, step through them by highlighting one at a time with the "<" and ">" keys.

Up to sixteen "cursors"—vertical, repositionable lines—are available to assist in making simple measurements of the active sweep. Cursor text boxes display (optionally) time, amplitude and sample number, or delta values relative to a paired cursor. Configure these and other cursor options by double-clicking on a cursor to open the Cursor Properties dialog box. A number of measurements for the first two sets of cursors, and the sections of trace they bound, can be quickly sent to the Results window using the ✓ and ✓+ buttons in the top-left of the Analysis window.

Data File Index

The Data File Index (DFI) window is a file management tool that allows you to construct an index of data files. Data files can be grouped together in separate DFI files, and then sorted according to a wide range of parameters reported for each file. These options give you great flexibility in being able to rapidly organize and find files from particular types of experiments—especially valuable in managing large amounts of data from archival sources such as CD-ROMs. Create a Data File Index from File > New Data File Index.

Lab Book

The Lab Book window is a text editor for logging events that occur while Clampex is running, e.g. that a protocol is opened, or when the holding level is changed. Events can be automatically written to it (see Configure > Lab Book Options) or add your own comments with the Tools > Comment to Lab Book command or by typing directly in the Lab Book. Several of the tools within Clampex offer the option of writing values to the Lab Book, e.g. the Membrane Test.

There is always a Lab Book window open, called the System Lab Book. Copies of this can be saved to disk for editing or archiving elsewhere.

Membrane Test

Membrane Test is a sophisticated utility for monitoring a number of parameters during the three stages of a patch-clamp experiment:

- **Bath:** Electrode resistance in the bath before patching (formerly called “Seal Test”).
- **Patch:** Patch resistance, to assist you in forming a gigohm seal.
- **Cell:** Cell resistance and membrane capacitance.

You can switch between these three stages using the Stage buttons at the top of the dialog box. When you switch from one stage to another the current parameter values are recorded in the Lab Book.

A selection of the following electrode and cell membrane properties are reported, depending on the current stage:

- Total resistance, Rt
- Access resistance, Ra
- Membrane resistance, Rm
- Membrane capacitance, Cm
- Time constant, Tau
- Holding current, Hold

Results Window

The Results window contains a spreadsheet for the display of measurements derived from cursors 1 and 2, and 3 and 4, in the Analysis window. These measurements include time and amplitude minimums, maximums, deltas, average slope, mean and standard deviation. You can only select contiguous rows and columns to perform standard copy and paste operations.

Like the System Lab Book, a Results window is kept open whenever Clampex is running. There can be only one Results file open at any one time (though you can view it in more than one window if you use the Window > New Window command). The Results window can be saved as a separate file at any time, and opened into Clampfit if desired.

Scope Window

The Scope window displays digitized data in real time during data acquisition (both View Only and Record). You can optionally use the Acquire > View Only command to preview data without writing it to disk, and then Acquire > Write Last to save the data to a file.

Multiple Scope windows can be opened with Window > New Scope. This can be useful for viewing incoming data at various magnifications, or with different display options.

In Episodic-mode acquisition, when Statistics have been enabled (from the Edit Protocol > Statistics tab), cursors define search region and baseline boundaries, and significant data points of the statistics measurements are marked with symbols in the Scope window. For gap-free or event-detected modes, trigger thresholds (Edit Protocol > Trigger tab) are shown with adjustable horizontal markers.

The View > Store Sweep command preserves the last acquired sweep in Episodic and Oscilloscope modes, so you can easily compare a particular sweep to other data.

Statistics Window

Statistics windows are graph-format windows, with a time X axis and subwindows for each of the statistical measurements recorded. Measurements from up to eight different search regions (configurable within each sweep for high-speed oscilloscope and episodic stimulation acquisition modes) are color-coded within the separate subwindows. A pane in the right-hand side of the window shows the search region color legend, and reports the most recent data in numerical form.

When statistics are recorded in Clampex, these are drawn into the Online Statistics window.   This special Statistics window becomes active, accepting data, as soon as any statistics are generated. It continues to be active until Clampex is closed, recording all statistics measured. Once open, the Online Statistics window cannot be closed, though it can be minimized. It continues to accept data in either state. New subwindows are automatically added to the window for each new type of statistics measurement enabled.

Once activated, the Online Statistics window can be cleared of the data it contains and the X axis reset to begin at time zero (Edit > Clear Statistics). This also resets the number of subwindows, to those enabled in the currently loaded protocol. Before clearing, or at any other time, you can save the contents of the Online Statistics window into a standard statistics (STA) file. These files can then be opened into their own Statistics window (in both Clampex and Clampfit) with the File > Open Other > Statistics command.

Measurements are written to the Online Statistics window when:

- In episodic and oscilloscope modes, Shape Statistics is enabled (Edit Protocol > Statistics tab).
- In gap-free and event-triggered modes, Threshold-based Statistics is enabled (Edit Protocol > Trigger tab).
- In episodic stimulation mode, the epoch Resistance Test is enabled (Edit Protocol > Wave tab)
- Tools > Membrane Test is run.

TELEGRAPHS

For many amplifiers, Clampex can receive and incorporate a range of “telegraphed” amplifier settings. Depending on the type of amplifier you have, the variable gain, lowpass filter, and whole-cell capacitance compensation settings can be telegraphed, with AxoClamp 900A and MultiClamp amplifiers telegraphing amplifier mode and signal scale factors and units as well. Gain telegraphs automatically update the signal scaling of the input channel, based on changes to the amplifier's gain knob. Lowpass filter telegraphs and capacitance compensation telegraphs store these settings in the data file header. With the AxoClamp 900A and MultiClamp amplifiers, amplifier mode changes can be linked to sequencing keys, so that, for example, Clampex automatically loads a suitable protocol when you switch between voltage or current clamp modes. The scale factor and units telegraphs from the AxoClamp 900A and MultiClamp amplifiers almost entirely automate signal setup in Clampex, leaving you only to name the input signals.

Clampex telegraphs are configured in the Configure > Telegraphed Instrument dialog:

→ Select the digitizer input channel on which you receive the signal that the telegraphed information is relevant to.

→ Select the amplifier type, which determines the configuration options you are offered. Note that if the amplifier does not support telegraphs you can manually enter the information that would otherwise be telegraphed. To do this:

- Select "(Manual)" as the telegraphed instrument in the configuration dialog.

→ Enter the appropriate settings in the Configure > Lab Bench > Input Signals tab. The section of the tab where these entries are made reports telegraphed values in the case of normal telegraphing. Telegraphed filter, gain and capacitance compensation settings are also reported in the Real Time Controls.

LAB BENCH

When setting up Clampex for data acquisition, having first configured the digitizer (Chapter 3, *"Setup"*) you must configure input and output signals for it. This is done in the Lab Bench (Configure > Lab Bench).

Clampex allows you to define several different signals for each digitizer channel. For each signal you need to set units and scaling so that Clampex displays data properly corrected for the parameter being read. Then, when you set up an acquisition protocol, you select the appropriate channels and signals from the Input and Output tabs in the Acquire > Edit Protocol dialog box.

For the Analog IN #0 and Analog OUT #0 channels, Clampex has a number of predefined signals that are properly scaled for a variety of Axon Instruments amplifiers, but you are able to configure virtually any type of signal that you want, for any of the input or output channels.

If you are setting up your own signals, the Scale Factor Assistant helps calculate the correct scale factor. It asks a few basic questions, usually answered by reading the values from the front of the amplifier, and then computes the appropriate scale factor.

The Lab Bench has a tab each for input and output signals.

Setting Up Input Signals

Use the Input Signals tab to:

- Define units of measurement and any signal offsets.
- Define the scale factor by using the Scale Factor Assistant.
- Enable software filtering.
- Add additional gain or amplification to the signal before it is digitized.

If you configure a telegraphed instrument, amplifier settings such as output gain, filter frequency, and membrane capacitance values are also displayed.

Setting Up Output Signals

Use the Output Signals tab to:

- Define output signal units and scale factors (using the Scale Factor Assistant)
- And depending on the options you have selected in the Overrides dialog box (Configure > Overrides)
  - Set the holding level
  - Set digital OUT channels

OVERRIDES

The Overrides dialog box (Configure > Overrides) gives you the option of switching the control of various parameters, normally set in the Protocol Editor, to other locations. Of most immediate interest here are the options for analog and digital holding levels. If the Overrides options for these are left unchecked, they are defined, protocol by protocol, in the Outputs tab of the Protocol Editor. If checked, then general levels are set from the Outputs tab in the Lab Bench. These levels then apply regardless of which protocol is being run. Note, however, that whichever location has control, immediate changes can be made to holding levels from the Real Time Controls.

HANDLING DATA

File Names

Before real data are recorded you should select a naming regime for new data files, and choose the default directory for these to be saved to. Both settings are made in the File > Set Data File Names dialog box. You can select date-based naming or choose a user-defined prefix, and both options have long and short forms.

By default, data files are saved per user in My Documents\Molecular Devices\pCLAMP\Data and protocols are saved per user in My Documents\Molecular Devices\pCLAMP\Params.

After completing an experiment, you can inspect and edit the data-selecting File > Last Recording opens the most recently saved file in an Analysis window. The file opens using the display defaults which have been selected in the File > Open Data menu item.

Save As

The original data file from a recording contains all data from all signals configured in the protocol under which it was recorded. You can save selected sections from such a file, defining a region to save with the cursors, and selecting just those sweeps and signals that you want included. First hide the signals you do not want, on the Show/Hide tab in the View > Window Properties dialog box, and then hide sweeps you do not want, using the View > Select Sweeps command. Select File > Save As and then press the Options button if the current options reported in the Save As dialog box are not what you want.

PROTOCOL EDITOR

The Protocol Editor is of central importance to Clampex, as it is where the greater part of experimental configuration occurs. Opened with either of Acquire > New Protocol or Edit Protocol, it has options for setting up all the various aspects of data acquisition: the acquisition mode, trial length or hierarchy, sampling interval, the channels and signals that will be used, the shape of the command waveform, and whether or not any triggering, statistics measurements, leak subtraction, pre-sweep trains, or math channels are used.

When a complete range of acquisition settings has been configured, this can be saved together as a protocol file (Acquire > Save Protocol As), and reopened later (Acquire > Open Protocol) as needed.

Setting the Acquisition Mode

Once the hardware is set up and signals have been configured in the Lab Bench, the first act in setting up an experiment is to choose the acquisition mode (see Data Acquisition Modes, page 16). This must be done first as many of the options in the Protocol Editor change depending on the mode selected. Selecting the acquisition mode is a simple matter of clicking one of the option buttons at the top of the Mode/Rate tab—the first tab in the Protocol Editor.

Trial Length

For acquisition modes other than Episodic stimulation you must select options for the length of the trial. This is done on the Mode/Rate tab. One option is to record data for an unspecified amount of time, i.e. until all available disk space is used, or until the data file reaches a 4 GB limit. Alternatively, you can set a fixed trial duration.

To help settle these options the amount of free space on the hard disk is reported on the tab, both in megabytes and in terms of the amount of recording time this gives you at the current settings. The time available is inversely related to the number of channels sampled and the sampling rate, so, if disk space is an issue, one way to free some up is to decrease the sampling rate. Another way to save disk space, if feasible, is to use fixed length or variable length acquisition rather than gap-free acquisition.

Trial Hierarchy

When Episodic stimulation is selected the Mode/Rate tab has options for the trial hierarchy. You need to enter the sweep duration, the number of sweeps in a run, and the number of runs in each trial. If you intend having more than one input channel, the sweep duration that you enter is the same for each signal.

If you are keeping an eye on disk space, the file size—for files created under the protocol at its current configuration—is reported beside the Sweeps/Run list box. Note that the number of runs per trial does not affect the file size—each run in a trial is averaged and only one final, averaged run is saved. The total amount of free disk space is reported at the bottom of the tab, both in megabytes and as the number of sweeps.

A breakdown of the sweep into First and Last Holding periods, and Epochs, is also provided. This refers to the command waveform, where the First and Last Holding periods are each automatically calculated as 1/64th of the specified sweep duration, during which time only the holding level is output. The command waveform is simultaneously output while recording data according to the hierarchy you configure here. The waveform is produced in each sweep, and the “Epochs” value reported here is the range of time available for you to configure its shape.

Sampling Rate

A trial’s sampling rate is also set in the Mode/Rate tab of the Protocol Editor.

The sampling rate is set per signal, so even if you acquire multiple signals, each signal has the displayed sampling rate.

The total throughput of the data acquisition system, i.e. of [sampling rate] x [number of signals] is displayed beneath the sampling interval field for nonepisodic modes, and at the bottom-right of the tab for episodic stimulation. This sampling rate is also set per signal in the generation of the command waveform.

pCLAMP supports split-clock acquisition, meaning that you can define two different sampling rates: a Fast rate and a Slow rate. You specify when to use these rates in the Sample rate row on the Waveform tab.

Start-to-Start Intervals

In Episodic stimulation mode (again on the Mode/Rate tab) the Start-to-Start Interval allows you to time the start of each sweep and/or run relative to the start of the one before it. This means that a Start-to-Start Interval must be equal to or longer than the sweep length or the total time of a run. Pre-sweep trains, P/N subtraction, and Membrane Teset Between Sweeps are included in calculating the sweep length. If you just want the computer to execute sweeps or runs as quickly as possible, then use the “Minimum” setting. A minimum setting results in no lost data between sweeps, producing a continuous record divided into sweeps. Note that if triggers are used to start a sweep, then the Start-to-Start Intervals are disabled, because timing is now controlled externally.

Averaging

Also located on the Mode/Rate tab, Averaging is relevant when more than one run has been selected per trial in Episodic stimulation mode. It is also offered as an option in High-speed oscilloscope mode.

Two types of averaging options are available: Cumulative and Most Recent. In Cumulative averaging, each run contributes the same weighting into the average. As runs are accumulated, each successive run contributes a smaller percentage to the average, which becomes increasingly resistant to change. This option is recommended for data with fairly stable baselines.

In Most Recent averaging, only the last *N* runs are used to calculate the average, so the average is sensitive to changes in the data. This is recommended for data with significant baseline drift.

Up to 10,000 runs can be averaged. The average is updated with every run, but changes to the average can be removed allowing you to revert to an earlier saved average. Thus, if bad data are received midway through a trial you can still save some results at least, by reverting to the last average before the undesirable data were recorded. Options for this are set in the Undo File section of the Averaging Options dialog box.

The Inputs and Outputs Tabs

The Inputs tab in the Protocol Editor is where you select the number of input channels for receiving data, and also where you select the signals for each of these channels. These are the signals that were configured in the Configure > Lab Bench Inputs tab.

On the Outputs tab you similarly select signals for the output channels. You are also able to set both analog and digital holding levels on this tab. The levels you set here are specific to the protocol you are configuring. If the holding level fields simply report their values and cannot be adjusted, that means that you have set the Configure > Overrides options to give control of these parameters to the Lab Bench.

In setting analog holding levels, remember that levels set in Clampex are in addition to those set on the amplifier. Many users have their amplifier holding level control turned off and control this entirely from Clampex.

Triggers

Data acquisition triggering is set in a protocol’s Trigger tab.

Your choice in the “Start trial with” list box determines the way that you tell Clampex you are ready for data viewing or recording. This is always initiated in the first case by selecting Acquire > Record or Acquire > View Only (or pressing their buttons in the Acquisition

Toolbar). Thereafter, the option you selected in “Start trial with” takes effect. The trial can be set to start immediately, or on receipt of some external or keyboard signal.

Once a trial has been started, Clampex can be set to await a trigger before any data are recorded. In modes other than Episodic stimulation, you can select a particular input signal and then set a threshold level for it. Rather than selecting a signal by name, you may want to go straight to the first signal enabled on the Input tab, with the “First Acquired Signal” option.

The “Pretrigger length” field allows you to set how much of the trace, before a threshold crossing, you want recorded. This setting is also used for the period that recording continues at the end of an event in Variable-length events mode. To avoid noisy signals causing false triggers you can refine trigger settings in the Hysteresis dialog.

In Episodic stimulation mode, the “Trigger source” option of “Internal Timer” causes the runs and sweeps in the trial hierarchy to be controlled by the settings in the Start-to-Start Interval section of the Mode/Rate tab.

Users can also enable a digital trigger signal from this tab. This uses the ADC CLOCK OUTPUT rear panel BNC on the Digidata 1320A/1321A digitizer, the TRIGGER OUTPUT rear panel BNC on the Digidata 1322A digitizer, or the SCOPE front panel BNC on the Digidata 1440A digitizer to send a trigger to devices such as an oscilloscope. When enabled, the port outputs a 5 V TTL signal to coincide with data acquisition.

If you are in an acquisition mode other than episodic stimulation, the trigger remains active for the duration of the time the signal remains over (or under, when you have selected negative polarity) the threshold level, selected in the Trigger Settings or Statistics Settings section immediately below the checkbox. Otherwise, in episodic stimulation mode, this trigger is always output, and is held active from the start of the first holding period in each sweep until the start of the last holding period in each sweep.

Threshold-Based Statistics

When acquiring in gap-free or event-detected modes, you can monitor measurements such as the event frequency or percentage of time above the threshold level. These threshold-based statistics are set up in a protocol’s Trigger tab, and use the same setting as for the Trigger Source.

When threshold-based statistics are enabled, the statistical data are recorded in, and can be saved from, the Online Statistics window. You can also choose to have the data automatically saved in a statistics file when a recording finishes.

Threshold-based statistics can serve as a handy indicator of the progress of an experiment. For example, it might be expected that in the presence of a certain drug, the channel activity will increase. If so, the percentage of time spent above threshold is a measure of this increased activity, and can be used as a criterion for continuing the experiment.

Statistics

Shape statistics, which measure various parameters of evoked events such as peaks, areas, slopes and rise times, are set in the Statistics tab of the Protocol Editor. They are available in episodic and oscilloscope modes.

Shape statistics can be measured from any of the input signals. Searches for different measurements can be configured for up to eight different search regions within the sweep.

Once you start digitizing data, you see vertical cursor lines bounding the search regions in the Scope window. These are numbered according to search region, and the region boundaries can be reset during data acquisition by simply dragging the cursors to new positions.

Shape statistics are written to the Online Statistics window, with options to automatically save the data from each trial to their own statistics file, and to clear the Statistics window after each trial.

Math Signals

Two analog input signals can be arithmetically manipulated and displayed as a separate online signal in the Scope window. This is set up in the Math tab of the Protocol Editor. Two signals can be scaled, offset, and arithmetically combined before being displayed, by using the General purpose equation. Or, the Ratio dyes equation can be used to measure cellular dye concentrations by ratioing the fluorescence signals from two photomultiplier tubes.

Waveform

You can define analog and digital waveforms in the Waveform tab of the Protocol Editor. These command waveform outputs are available only in Episodic stimulation mode.

Up to four analog stimulus waveforms can be generated simultaneously, one for each of the four output channels tabs at the bottom of the dialog. Simultaneously, eight digital outputs can be enabled on one of the output channels. Epoch-driven waveforms are defined in an Epoch Description table. This consists of up to ten epochs (i.e. sections of the sweep) preceded and followed by a holding-level period that is predefined as 1/64 in duration of the sweep length. For analog waveforms, each of the epochs can be configured to step to and hold a particular voltage, to increase or decrease linearly in a ramp, or to produce a rectangular, triangular, sinusoidal or biphasic wave train. Click in the Type row for the epoch you want to configure, to assign one of these options. Amplitudes and durations of each epoch can be systematically increased or decreased by setting delta values, i.e. fixed amounts of change in a parameter with every sweep.

Train outputs can be configured for both analog and digital outputs. Trains can be generated as square pulses, sawtooth pulses, biphasic pulses and a cosine wave. A digital bit pattern can be specified with binary numbers (0, 1), and individual digital bits are enabled for trains by using the * symbol.

For more flexible control of the analog waveform, see *"User List"* on page 43. If the epoch-based analog waveform is still not flexible enough for your needs, the Stimulus File option can be used to read data from an ABF or ATF file and output it as an analog waveform.

While you are building the command waveform, you can see how it looks by pressing the Update Preview button in the bottom-right of the Protocol Editor. This opens an Analysis window displaying the waveform. You can keep this window open while you make changes in the Epoch Description table, pressing Update Preview again to see the latest changes made.

Pre-sweep Trains

A train of pulses can be used to "condition" a cell prior to the main stimulus waveform. This is configured in the Stimulus tabs of the Protocol Editor.

The Clampex pre-sweep train outputs analog pulses composed of repeated baseline and step levels. After the pre-sweep train is completed, the output can be held at a post-train level. If the number of pulses in the train is set to zero, only the post-train level is generated, which is a convenient way to change the holding level for a specified duration before each sweep.

The pre-sweep train can be output on either the same analog out channel as the stimulus waveform, or on the other analog out channel. Data are not acquired during the pre-sweep train period, so you cannot observe its effects in Clampex. To view these, run AxoScope concurrently in gap-free mode with a MiniDigi digitizer.

P/N Leak Subtraction

Leak subtraction is a technique typically used in voltage-clamp experiments to correct for a cell's passive membrane current, i.e. the leakage current. It is configured in the Protocol Editor's Stimulus tabs.

In P/N Subtraction, a series of scaled-down versions of the command waveform is generated, and the responses are measured, accumulated, and subtracted from the data. Scaled-down versions of the waveform are used to prevent active currents from being generated by the cell. The number of these scaled-down waveforms, entered as the number of subsweeps, is the "N" referred to in the name of this technique. The waveform (pulse P) is scaled down by a factor of 1/N, and this is applied N times to the cell. Since leakage current has a linear response, the accumulated responses of the subsweeps approximates the leakage current for the actual waveform. This is subtracted from the input when the actual waveform is run.

For added flexibility in preventing the occurrence of active currents, the polarity of the P/N waveform can be reversed. In this case, the P/N accumulated response is added to the sweep of data. Also, to prevent a conditioning response from the cell, the P/N waveforms can be issued after the main stimulus waveform.

In Clampex 10, both the raw data and the P/N corrected data are saved. During data acquisition, only one of these is displayed.

User List

The User List, on each of the Stimulus tabs, provides a way of customizing one of a range of analog and digital output features, overriding the generalized settings made elsewhere in the Protocol Editor. The selected parameter can be set to arbitrary values for each sweep in a run by entering the desired values in the List of parameter values field.

So, for example, rather than having sweep start-to-start times remain constant for every sweep in a run, specific sweep start-to-start times can be set for each sweep. Alternatively, rather than being forced to increase or decrease the duration of a waveform epoch in regular steps with each successive sweep (by setting a duration delta value), you can set independent epoch durations for each sweep. All aspects of the conditioning train, the number of subsweeps in P/N Subtraction, and command waveform amplitudes and durations (besides other output features) can be overridden from the User List.

DATA ACQUISITION

Once you are ready to receive data, there are a number of options for how you go about it. These are contained in the Acquire menu, or you can use the corresponding toolbuttons.

You can digitize data and view the result in the Scope window, without saving anything to disk, with Acquire > View Only. Once the acquisition completes, you can save it to disk by selecting Write Last. Alternatively, you can simply press the Record button and the data are saved to disk during the acquisition. You can overwrite the last recorded data file by selecting the Rerecord button. You can also repeatedly perform the protocol, in either View Only or Record modes, by pressing the Repeat button.

REAL TIME CONTROLS

The Real Time Controls panel allows you to monitor the status of the experiment and easily control several selected input and output parameters while digitizing data. You can quickly test different experimental settings when viewing live data without having to open the Protocol Editor or Lab Bench—settings such as:

- Changing the holding levels of cells
- Controlling the digital outputs to a perfusion system
- Adjusting the sampling rate
- Applying filtering to the data signal currently selected.

The panel also reports where you are in an acquisition, in terms of elapsed time and sweep and run, and whether or not you have a conditioning train or P/N Subtraction enabled.

To display the Real Time Controls panel, choose View > Real Time Controls. By default, it appears along the left frame of the main window. To make it a floating window, drag it from the frame and position it like a standard dialog box.

The following rules apply to the parameters available in the Real Time Controls:

- All parameters are available when acquiring data in View mode (Acquire > View Only). Use this mode to experiment with sampling rates and filter settings.
- A few parameters are available when recording data in nonepisodic modes. Only analog and digital output levels can be changed, and a comment tag is inserted into the data at each change.
- Most parameters are disabled when recording data in episodic mode. Only digital output levels can be changed, and a comment tag is inserted into the data at each change.

SEAL AND CELL QUALITY: MEMBRANE TEST

Tools > Membrane Test is a sophisticated set of controls for monitoring the electrode, the seal and the membrane, during and after the patching process. It calculates a range of accurate measurements by generating a square pulse of known dimensions and measuring the response signal.

The Membrane Test has 3 stages, each with its own set of parameters:

- Bath stage monitors the electrode in the bath solution.
- Patch stage monitors the creation of a gigohm seal.
- Cell stage monitors cell membrane resistance and capacitance.

The following values are calculated:

- Total resistance, Rt
- Access resistance, Ra
- Membrane resistance, Rm
- Membrane capacitance, Cm
- Time constant, Tau
- Holding current, Hold

Separate “Play”, “Pause” and “Stop” buttons allow you to control when the Membrane Test is running, as well as its settings and test history. Membrane Test can control the holding potential and stop generation of the test pulse. You can also configure and trigger a pulse train to interrupt the test pulse. Before using Membrane Test you must first define input and output signals through Configure > Membrane Test Setup.

Membrane Test measurements are displayed numerically in the Membrane Test window, and can also be charted in the Online Statistics window. You can automatically save the test measurements as an STA file with other statistics measurements from the Online Statistics window. A snapshot of the Measurements at the current time point can also be saved to the Lab Book.

Membrane Test can also be run automatically between every sweep. Configure this on the Acquire > Edit Protocol > Stimulus tab. In this mode the Membrane Test data are written out to the Online Statistics window and displayed in the Real Time Controls. The settings in Configure > Membrane Test Setup are also used in this mode. The Membrane Test dialog can be displayed, but it is not used when calculating Membrane Test Between Sweeps.

The Membrane Test pulse train is used to deliver a series of stimuli without closing the Membrane Test (the test pulse ceases while the train is run). The pulse train can be configured to use a different signal from the primary test pulse, possibly on a different output channel, so that you can send it to a separate stimulating electrode.

In order to calculate many of the measurements, the transient current response is fit by a single exponential. From this the decay time constant is calculated and used to ensure the signal reaches a steady state for reliable current measurements. Similarly, the fitted curve is used to calculate the transient peak, from which the access resistance is derived. The options in Configure > Membrane Test Setup let you define how much of the falling transient from each pulse edge to use for curve fitting. This same dialog also allows you to choose the timing and amplitude of the pulse train.

In addition to measuring important initial seal and cellular parameters, you can use the Membrane Test to perform an experiment. For example, you can use the pulse train to depolarize and stimulate a cell, and induce exocytosis in a secretory cell. Then, you can record the secretory cell's whole-cell response and monitor relative capacitive changes.

You can also include the Membrane Test in sequencing key series, for example to monitor seal or access resistance as part of a set sequence of steps

For the algorithms used to calculate Membrane Test values, see *"Membrane Test"* on page 163 (in Chapter 10, *"pCLAMP Analyses"*). See also *"Membrane Test Tutorial"* on page 77 (in Chapter 5, *"Clampex Tutorials"*).

Membrane Test Measurements

In pCLAMP, the resistance due to an electrode alone is termed the electrode resistance (R_e). This is also sometimes termed the pipette resistance (R_p). Usually, in addition to electrode resistance, there is a degree of resistance due to largely unknown environmental factors near the tip of the electrode. Cellular debris, air bubbles and poorly conducting solution might all contribute to this additional resistance (which we will call R_{debris}).

The sum of the electrode resistance and resistance due to these additional factors is the access resistance (R_a):

$$R_a = R_e + R_{debris}$$

Access resistance is also commonly termed series resistance (R_s). While R_s is the term used with Axon Instruments amplifiers, we avoid its use in pCLAMP as it can be confused with seal resistance, dealt with below.

Resistance across the cell membrane is called membrane resistance (R_m).

The resistance between the headstage and ground is the total resistance (R_t). Strictly speaking, when the electrode is in contact with the cell, there are two pathways from the electrode tip to ground—one traversing the cell membrane and the other bypassing it, leaking directly from the tip into the bath. However, the comparative resistances in the two pathways are such that we can generally ignore one or the other pathway, depending on whether the electrode has access to the cell interior or is patched to the outer surface. The two pathways are illustrated in Figure 4.1.

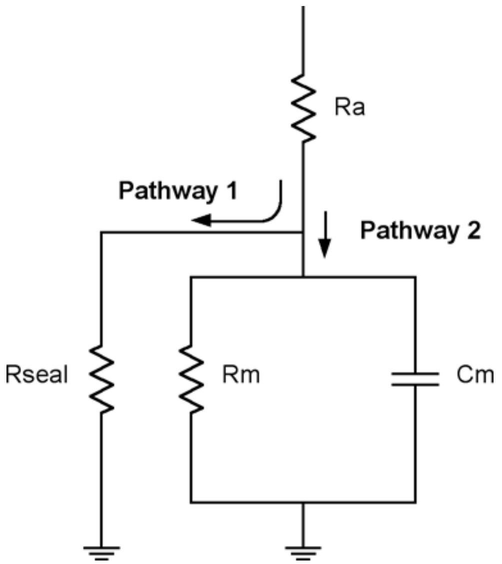


Figure 4.1: Idealized circuit showing two pathways from electrode tip to ground.

Patch Scenario

When a seal is created but the cell membrane is not ruptured, the resistance of the minute section of membrane in the patch is very high, so any current is likely to leak through the membrane/electrode seal. In Figure 4.1, this is equivalent to Pathway 2 being removed and all current taking Pathway 1. The resistance to this leakage, determined by the quality of the seal attained, is generally termed the seal resistance. In this case total resistance effectively consists of the access resistance and the seal resistance in series:

$$R_t = R_a + seal\ resistance$$

In any successful seal, the seal resistance is orders of magnitude larger than the access resistance so that:

$$R_t \approx seal\ resistance$$

Ruptured Patch Scenario

For a ruptured patch, when a satisfactory GΩ seal has been achieved, we hope that most current follows Pathway 2. In fact, with R_m and R_{seal} in parallel, we cannot distinguish these from each other. In this case, then, we assume all resistance is due to R_m. Under this assumption the total resistance consists of the access resistance and membrane resistance in series:

$$R_t = R_a + R_m$$

Terminology

As noted above, seal resistance is sometimes represented "R_s", which is also used to indicate series resistance (i.e. access resistance). For this reason we avoid the use of "R_s" in pCLAMP, favoring instead "R_a", "R_t" or the words "seal resistance" or "R_{seal}" when we need to refer to seal resistance specifically.

Bath

When the electrode is in the bath the total resistance will be just the electrode resistance (R_e). As the electrode approaches the cell, debris around the electrode tip may add further resistance (R_{debris}).

Seal

Only once a tight seal is formed with the cell membrane is the reported value effectively a measure of seal resistance (R_{seal}).

Membrane Test allows you to control the holding potential—a feature that is often used to hyperpolarize the cell to aid in seal formation. This control remains only for as long as the Membrane Test is run.

TIME, COMMENT, AND VOICE TAGS

Time Tags, Comment Tags, and Voice Tags let you annotate data while it is being collected. Each of these tags is available through the Acquire command during data acquisition, as well as from dedicated toolbuttons.

Time tags insert a simple numerical time-stamped tag into the data file. A Comment tag allows you to add an additional line of text to each tag. It is important to note that a comment tag is inserted when the tag is activated, not when you finish typing the comment. Clampex keeps a list of comment tags that have been used so that you can quickly recall a previous tag rather than retyping it.

If the computer includes a sound card, you can insert Voice (audio) tags into data files. This is analogous to recording your voice on an audio channel when saving data to a VCR tape. To configure Voice tags, choose Configure > Voice Tags. When you open the data file, double-click on a Voice tag to hear the audio comment.

JUNCTION POTENTIAL CALCULATOR

The Junction Potential Calculator provides a wealth of information on the measurement and determination of junction potentials created by solution changes. To start the calculation, choose Tools > Junction Potential. The Calculate Junction Potentials dialog box opens and lets you define the experiment type and temperature, and the type of ground electrode being used.

After providing this information, the Junction Potential Calculator graphically displays the various elements of the junction potentials. To determine the new junction potential, input the concentrations of the various ions in the new solution. The Junction Potential Calculator computes the junction potential that will be created by the solution. You can choose the ions from a supplied ion library list, or add your own ion definitions to the list. Additionally, you can copy the results of the Junction Potential Calculator to the Lab Book, save them to disk, or print them.

For more detailed information on how to use the Junction Potentials command, refer to the online Help.

CALIBRATION WIZARD

The Calibration Wizard lets you define the scaling and offset of a signal, after the data has been acquired. This is analogous to drawing a known scale bar on chart paper when recording continuous data to a pen recorder, and then measuring the scale bar afterwards to determine the scale factor for each signal of data. The Calibration Wizard is available when viewing data in the Analysis Window, and is activated by choosing the Tools > Calibration Wizard command.

When using the Calibration Wizard, you have the option of setting the scale factor based on measurements you make from the data file, or applying a known scale factor to the data. If you choose to set the scale factor based on the data file, you simply position Cursors 1 and 2 over two regions of known amplitude and indicate their values. The Calibration Wizard then computes the appropriate scale factor and offset, and optionally allows you to apply these values to the Lab Bench, so that any other data files that use the same signal also have the correct scale factor and offset.

Alternatively, if you already have defined the scale factor and offset in the Lab Bench, you can directly update these settings in the data file using the Calibration Wizard.

Existing data files can also be rescaled in Clampfit using Edit > Modify Signal Parameters.

SEQUENCING KEYS

The Sequencing Keys command in the Configure menu lets you associate events, or a sequence of events, with a keystroke action.

For example, you can define a single key to flip a digital OUT high to activate a solution change, and another key to flip the digital OUT low to terminate the solution change. These keys are available as toolbar buttons in the Sequencing toolbar. You can also define a tooltip (a popup message) that reminds you what event is associated with each button.

In addition to setting various digital OUTs, the sequencing keys also let you change the holding levels, insert a comment tag, start a Membrane Test, load and run a protocol, or display a prompt.

In addition to associating a keystroke with an event, you can use the sequencing keys to link one event to another, and run an experiment in an automated fashion

For example, you may want to start a Membrane Test, and when it finishes, run an I-V protocol, perform a solution change, and then run the I-V protocol again. Using the Sequencing Keys, you can link these events together and define the timing interval between them. You can save an entire sequencing series to disk, and maintain several sets of sequences for each type of experiment that you perform.

LTP ASSISTANT

Long-term potentiation (LTP) and long-term depression (LTD) are terms for a broad range of experiments designed to investigate synaptic plasticity. In particular, it has been found that certain stimulus regimes alter synapse function, both to increase (LTP), and decrease (LTD) postsynaptic response. The LTP Assistant in the Tools menu provides a convenient interface for the setup of LTP and LTD experiments. This section provides an in-depth discussion of the LTP Assistant's organization and functionality.

The LTP Assistant brings together a range of Clampex functions, presenting only the options most commonly needed for LTP and LTD experiments, and formatted in a way that follows a natural sequence for the setup of this type of experiment.

The LTP Assistant should be of use to both new and experienced LTP experimenters, however it requires a basic knowledge of Clampex (for example, the Lab Bench). A range of preconfigured protocols are included along with default stimulation waveforms. Users can select the default protocol nearest to their needs and adjust its settings. Alternatively, they may have protocol files already saved which they can copy into the LTP Assistant.

For those wanting to design experiments with configuration options beyond those offered in the LTP Assistant, it can still be used for overall experiment management. For example, if the range of protocol definition options provided within the LTP Assistant is insufficient, users can open the protocol editor from within the LTP Assistant and define their protocol with the broader range of options offered there.

The LTP Assistant offers the following features:

- Creation and sequencing of the baseline and conditioning stages of an experiment.
- Configuration of the stimulus waveforms used during baseline and conditioning stages, for both digital and analog stimulation.
- Conditioning pulse trains of indeterminate length.
- Alternation of presynaptic stimulus between two pathways.
- Postsynaptic command stimulus to coincide with or follow presynaptic conditioning stimulus.
- Preconfigured statistics measurement.

The assistant has four tabs, used as follows:

- **Sequencing:** General configuration of the stages that make up the experiment.
- **Inputs/Outputs:** Configuration of Clampex for the input and output channel and signal connections in the experiment setup.
- **Baseline:** Configuration of the protocol used in baseline stages of the experiment. This defines the baseline stimulation waveform. Enabling of default statistics measurements is from this tab.
- **Conditioning:** Configuration of the protocols used in conditioning stages of the experiment. This includes conditioning stimulation waveforms and enabling and configuration of a paired postsynaptic command waveform.

Experiment and Data Organization

The set of configuration options defined within the LTP Assistant is called an “experiment”. Experiments are saved from within the Sequencing tab. This creates a folder containing a sequencing key file (with the experiment’s name and *.sks extension) and as many protocol files (*.pro extensions) as are named in the Sequencing tab, i.e. are incorporated into the experiment. These constituent files are all saved at the same time as an experiment. Experiment folders are saved in My Documents\Molecular Devices\pCLAMP\Params\LTP1\.

Protocols can be reused for different experiments, but in each case are copied to the relevant experiment folder. As each copy is placed into a different folder from the original file you can keep its original name, or rename it if you want, on the Sequencing tab.

To run an experiment:

- Open the experiment in the LTP Assistant.
- Close the LTP Assistant. This loads the experiment’s sequencing key file into Clampex.
- Start the experiment by pressing the sequencing key for the first stage.

- Subsequent stages can also be started with sequencing keys, or might have been configured to run automatically when the previous stage finishes.

Each run of a protocol results in one data file (if you have chosen to record it). The data files generated during an experiment are named and located according to standard Clampex functionality, set in File > Set Data File Names. You can, if you want, later concatenate the files from an experiment in Clampfit, with Analyze > Concatenate Files, provided they were recorded under similar conditions.

As normal for Clampex, all statistics are written continuously to the Online Statistics window. You should, then, clear the previously logged statistics (Edit > Clear Statistics) before you start an experiment. If the final (baseline) protocol in an experiment runs to completion, the statistics for the whole experiment are automatically saved in one file by default. If, on the other hand, you end the experiment manually, you need to manually save the statistics (File > Save As when the Online Statistics window is selected).

Signals and Channels

LTP and LTD experiments involve, at minimum, one source of presynaptic stimulation and one postsynaptic recording electrode. Stimulation might be delivered extracellularly, with a stimulus isolation unit (SIU), or intracellularly, with a standard amplifier electrode. The postsynaptic recording electrode can be intracellular or extracellular.

Beyond this, and depending on the number of amplifiers available to the experimenter (and the number of electrodes these support), many more configurations are possible. The LTP Assistant aims to streamline software configuration for the most common electrode placements.

Output Channels

The LTP Assistant has the capacity to enable and configure four output channels:

- Two analog outputs (Analog OUT #0 and #1)
- Two digital outputs (Digital OUT #0 and #1)

It is anticipated that the analog waveforms are output to standard amplifier electrodes, while the digital outputs drive SIUs.

The output channels are grouped on the Inputs/Outputs tab to allow configuration of one or two “pathways”— routes of neuronal activity across a synapse from a presynaptic to a postsynaptic cell. Thus, you can enable a maximum of two output channels for presynaptic stimulation. If you have enabled two digital presynaptic outputs, any further outputs (necessarily analog) can only be for postsynaptic electrodes, assumed to be in the same pathways.

The digital outputs can only be used for presynaptic stimulation. Given the potential for two pathways, it is possible to have one digital and one analog presynaptic command, which would then leave one analog channel for postsynaptic stimulation. When two presynaptic outputs are enabled, you can also choose to alternate the baseline protocol delivery between these, sweep by sweep. This is enabled from the Baseline tab.

Presynaptic and postsynaptic analog output channels have different waveform configuration options. Presynaptic channels can be configured to deliver pulses or pulse trains. Postsynaptic channels are restricted to delivering postsynaptic pairing steps or pulses (for conditioning stages).

Note that, even if they have not been selected for presynaptic or postsynaptic commands, the analog output channels are always active, and always have a signal and holding level assigned to them. In this case, they can be used to clamp the voltage on a recording electrode, and you need simply ensure (on the Inputs/Outputs tab) that you have the correct signal and holding level for the channel, and connect it to the recording electrode. In general, it is preferable to clamp recording electrodes from within the LTP Assistant rather than with the amplifier, as the holding level is then written into the data file for later reference.

Input Channels

Four analog input channels (Analog IN #0, #1, #2, and #3) are available for recording or viewing data.

The LTP Assistant supports two analog output channels, so it is not possible to use the LTP Assistant to clamp four recording electrodes. In such a case, two of the recording electrodes can be clamped from within the LTP Assistant, and the two remaining electrodes would then need to be clamped by the amplifier holding level control.

Alternatively, the experiment protocols can be manually setup from within the protocol editor to control four analog output channels, and then configured for sequencing via sequencing keys.

Signals

Selection of signals for input and output analog channels follows standard Clampex procedure. Signals must be created and configured in the Lab Bench, in association with particular digitizer channels. Then, in the LTP Assistant Inputs/Outputs tab, select specific signals for the channels you have enabled—this is the same as in the Input and Output tabs of the protocol editor for standard protocols (if you are unfamiliar with the relation between signals and channels in Clampex, see *"Definitions"* on page 7).

Example Setup

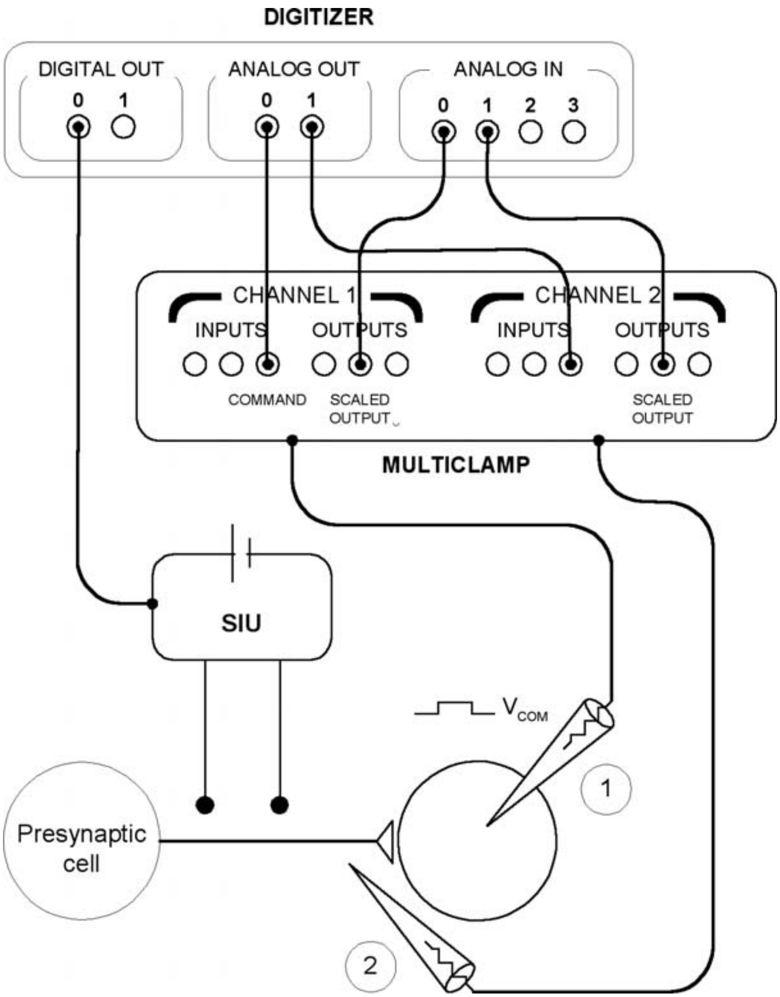


Figure 4.2: Example of LTP experiment setup, with digital presynaptic stimulation, postsynaptic intracellular voltage step command and current recording, and extracellular recording.

This setup has the following features:

- Presynaptic stimulation is delivered by an SIU via digital OUT #0.
- Postsynaptic intracellular electrode #1 is clamped via analog OUT #0, and can be configured to deliver a paired voltage step during conditioning stages. The electrode records to analog IN #0.
- Electrode 2 records extracellular responses to analog IN #1. It receives no command waveform, but is clamped via analog OUT #1.

Temporal Structure of LTP Experiments

LTP and LTD experiments focus on changes in synaptic behavior. The paradigm experiment thus typically has three stages:

- **Baseline:** The presynaptic cell is stimulated at a low frequency and the postsynaptic cell monitored to establish the base behavior of the synapse under investigation. Raw postsynaptic response data may or may not be recorded, but typically statistics are recorded, e.g. peak amplitudes and rise slopes. The baseline stage is also referred to as the "test", or "test 1" period.
- **Conditioning:** A different stimulus is applied to the synapse to evoke LTP or LTD. This might involve one or more "trains" (or "bursts") of pulses delivered via the presynaptic cell, or some pattern of presynaptic stimulation accompanied by a command signal to the postsynaptic cell ("pairing"). This stage is sometimes called "induction", and sometimes "tetanus", though this latter term more correctly names a particular type of conditioning stimulus.
- **Baseline:** the postsynaptic cell is monitored under the same stimulus regime used in the first baseline stage, in order to detect changes brought about during conditioning. This second application of the baseline is generally run for longer than the first. Again, this stage is sometimes referred to as "test" or "test 2".

As the first step in creating a new experiment setup a default three-stage "baseline-conditioning-baseline" format like the one above is displayed in the Sequencing tab, to which you can add as many stages of either sort as you like.

Each stage of the experiment is identified as either a baseline or conditioning protocol, determining the range of waveform configuration options available for that stage. The conditioning protocol is further configured with default settings for tetanus, theta or LTD stimuli. You can change these settings later in the Baseline or Conditioning tabs. You are able to use a given protocol in any number of stages; in fact, this is enforced in the case of baseline stages:

- Only one baseline protocol can be used in an experiment.

So you can have as many baseline stages as you want, but they must use the same protocol (and hence have the same stimulus waveform). On the other hand, there is no limit to the number of conditioning protocols you can use in an experiment, if you have more than one conditioning stage.

As well as determining whether a stage is baseline or conditioning on the Sequencing tab, you select whether each stage will be viewed only or recorded to disk. Statistics measurements can be taken in either case.

Stage Timing and Sequencing

The LTP Assistant creates and displays sequencing keys in the sequential order in which they were created. However, this ordering of experiment stages does not control the order in which they are actually executed. Each stage is automatically assigned a "Start key", and a "Next key"—the Start key of the next stage to run. This allows you to have the stages run in any order you want, by changing the Next key assignment in a protocol's "Properties". So, you can configure the completion of one stage to automatically start the running of another stage, or manually interrupt a stage and start the next stage by pressing its sequencing key. You can also have a stage link to itself, so that it loops continuously until you manually start another stage.

The duration of each stage is set in the Sequencing tab. This can either be set to the acquisition time defined in the protocol Sweeps section, or you can override the protocol by setting a different time here. This setting can thus cut short a protocol's running time or, if longer than the protocol, hold the experiment in a state of inactivity until the next stage is scheduled to start. Baseline stages are set to run for 1000 sweeps by default. This generally gives you a longer protocol run-time than needed, so that it is likely that a steady baseline response is achieved, after which you can then manually trigger the conditioning stage. In contrast, conditioning stages are usually short and should generally be set to run for the protocol acquisition time, automatically calling the following baseline stage when finished.

Protocols

The LTP Assistant has one tab each for baseline and conditioning protocol configuration. These contain sub-sets of Clampex's protocol-definition options. If you want further configuration options or simply want to check any default settings not revealed in the LTP Assistant, press the Edit button at the bottom of the tab to view the currently loaded protocol in the main protocol editor.

Both tabs allow you to configure a pattern of rectangular pulses for presynaptic stimulation. The same set of options is offered for digital and analog waveforms, except that you must enter pulse amplitudes for analog signals (the same pulse amplitude is used throughout the waveform). The default sampling rate for both baseline and conditioning protocols (not revealed in the LTP Assistant), is 10 kHz.

In the "Sweeps" section, the start-to-start time determines the frequency at which a waveform is delivered. This cannot be less than the duration of the sweep length, which is a combination of the waveform length and derived pre- and postwaveform holding periods (each 1/64th of the waveform length). The waveform length must be of sufficient duration to contain the presynaptic waveform and (if enabled) the resistance test or postsynaptic pairing. You can view, record, and take peak measurements from the input signals while the waveform is being output (i.e. for the duration of the sweep length), but you must then wait until the start of the next sweep (determined by the sweep start-to-start time) before more data are digitized.

For baseline protocols only, if you have enabled two presynaptic stimulation outputs, you can alternate delivery of these, sweep by sweep. First, sweep 1 of the protocol configured for OUT #0 is delivered, while OUT #1 is held inactive (for digital outputs) or at holding level (for analog outputs). Next, sweep 2 of the protocol configured for OUT #1 is delivered, while OUT #0 is held inactive or at holding level. For the next sweep, sweep 3 on OUT #0 is delivered, and so on. It is possible to alternate between digital and analog outputs, and combinations of these.

Conditioning protocols do not allow alternation. If two presynaptic output channels are enabled, whether or not these have alternation enabled in the baseline stage, they are both available for concurrent delivery during conditioning stages, unless both outputs are digital, in which case channel OUT #0 is available for configuration, and OUT #1 is held inactive.

LTP Assistant differences between options for the baseline and conditioning stage protocols are summarized in the following table:

Table 4.1: Options for the baseline and conditioning stage protocols

	Baseline Protocols	Conditioning Protocols
Alternating sweeps	Alternate stimulation waveforms, sweep by sweep, between two presynaptic channels.	Not available.
Number of sweeps	1000 sweep default.	User configurable.
Pulse trains	Not available. Up to four pulses per sweep.	Up to four trains per sweep.
Statistics	Peak amplitude, time of peak and slope measurements.	Not available.
Postsynaptic pairing	Not available.	Step postsynaptic voltage or current for entire conditioning period, or in a pulse following presynaptic stimulus.

With the “Copy From” button on each protocol tab, you can select a protocol to copy into the LTP Assistant. Protocol settings are automatically changed, if necessary, to be consistent with the settings current in the assistant at the time, e.g. the signals and sampling rate are changed to those of the protocol already loaded in the LTP Assistant. Once copied into the experiment, you can adjust the protocol’s settings at will.

The Preview button in the LTP Assistant opens an Analysis window displaying the waveform you have configured. This window updates automatically as you make changes to waveform configuration within the LTP Assistant.

Presynaptic Stimulus Waveform Definition

Waveforms for both baseline and conditioning protocols are built up from “pulses”, with rapid sequences of pulses (“trains”) configurable for conditioning protocols only. A pulse is an output (current or voltage) step, defined in terms of its amplitude and duration.

As noted above, digital and analog waveform definition is essentially the same, differing only in that you must set a pulse amplitude (“pulse level”) for analog signals. You must enter an absolute value for this, rather than a value relative to the baseline.

Existing Clampex functionality dictates that the basic unit for waveform configuration is the sweep. Each sweep has ten configurable subunits: the epochs. Epochs are not referred to in the assistant, but their role in waveform definition can be seen, if you want, by opening the protocol editor and viewing the Waveform tabs. You are able to set the total duration of the epochs, and hence of the time available for waveform configuration, with the “Waveform length” field in each of the Baseline and Conditioning tabs. With one epoch reserved for an optional initial delay period at the start of the sweep, and the final two for a postsynaptic pairing delayed pulse (in conditioning protocols), seven remain for presynaptic waveform configuration. This means that you can have a maximum of four pulses per sweep for baseline protocols, or four trains per sweep for conditioning protocols. One or two pulses per sweep is adequate for baseline stimulation in most experiments, and, with the capacity to repeat sweeps, four trains per sweep is sufficient for many conditioning stimuli as well. If not, you can build additional conditioning stages into the experiment and string these together on the Sequencing tab, so that the total conditioning stimulation might consist of several conditioning stages.

Pulses and trains can be described in terms of a number of different parameters. The parameters used for sweep waveform configuration in the LTP Assistant are illustrated in Figure 4.3.

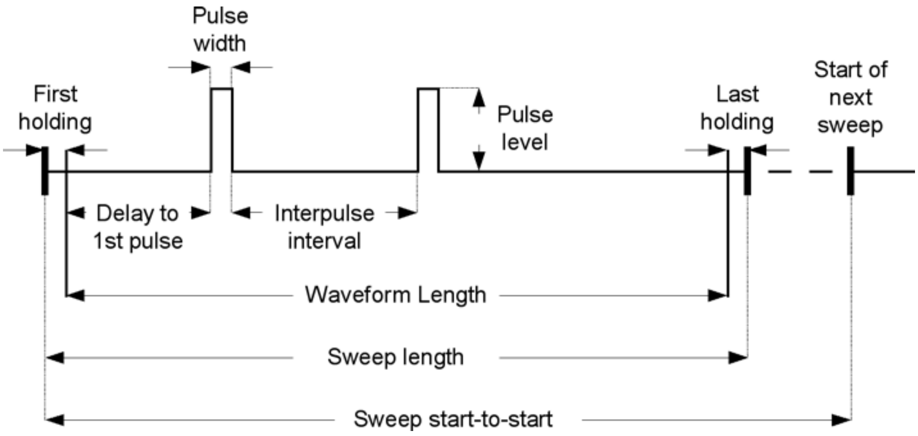


Figure 4.3: Baseline sweep with two pulses per sweep showing configuration parameters used on Baseline tab.

Note that the first and last holding periods are not referred to in the waveform configuration interface, but are shown in the diagrams to illustrate the difference between the value entered in the “Waveform length” field and the reported “Sweep length”.

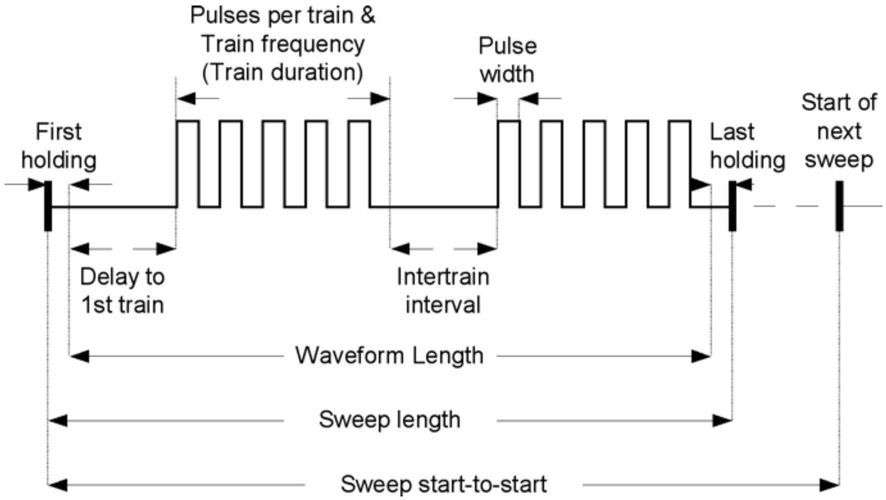


Figure 4.4: Conditioning sweep configuration with two trains per sweep, showing parameters used on Conditioning tab.

Note that because the pulses in a train each consist of a step and a following section at the holding level, trains similarly always begin with the start of a pulse, and end after a period at the holding level (equivalent to the interpulse interval within the train). The length of this period depends upon the pulse width and frequency you have set.

Predefined Protocols

Protocol definition does not start from a blank slate. When you first outline the overall structure of an experiment on the Sequencing tab, default protocol settings are loaded for each stage you create. One default protocol is used for all baseline stages, while you are given a choice of three default protocols for conditioning stages. This is simply to save you time, allowing you to select a starting point close to the configuration you want. The four default protocols have the following presynaptic waveform configuration.

Baseline: A single pulse of 1 ms at 0.05 Hz (20 s start-to-start interval). 51.6 ms of data is acquired for every pulse, starting 5.8 ms before the onset of the pulse.

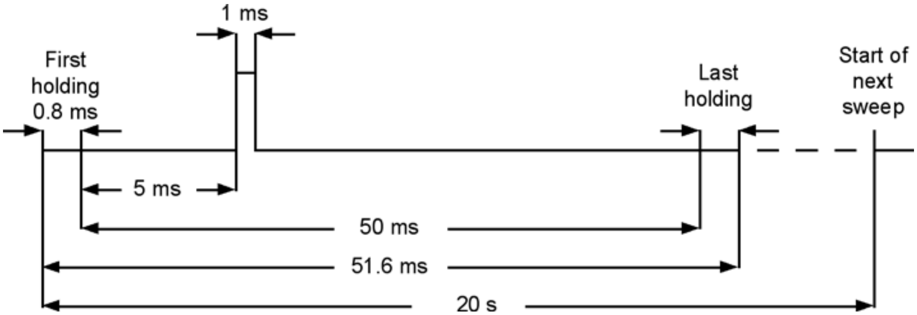


Figure 4.5: Default baseline presynaptic waveform.

- Sweep start-to-start: 20 s (0.05 Hz)
- Waveform length: 50 ms
- Sweep length: 51.6 ms
- Pulses per sweep: 1
- Pulse level: 20 mV
- Delay to first pulse: 5 ms
- Pulse width: 1 ms
- Interpulse interval: not applicable
- Sampling interval: 100 µs (10 kHz)
- No. of sweeps: 1000

Tetanus: This is the default conditioning protocol that each new experiment starts with, and is the protocol used if you add a new “Tetanus” conditioning stage from the Sequencing tab Add dialog. Four 100 Hz, 1 s trains are delivered 20 s apart.

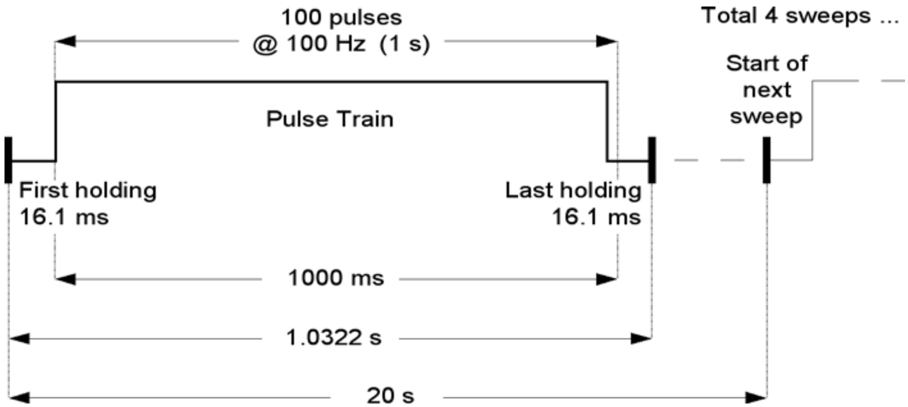


Figure 4.6: Default “Tetanus” conditioning stimulus.

- Number of sweeps: 4
- Sweep start-to-start: 20 s (0.05 Hz)
- Waveform length: 1000 ms
- Sweep length: 1.0322 s
- Trains per sweep: 1
- Pulse level: 20 mV
- Delay to first train: 0 ms
- Train duration: 1000 ms
- Intertrain interval: not applicable
- Pulses per train: 100
- Train frequency: 100 Hz
- Pulse width: 1 ms
- Sampling interval: 100 µs (10 kHz)

Theta: A series of ten short 100 Hz pulse trains, of five 1 ms pulses each, are delivered at a frequency of 5 Hz.

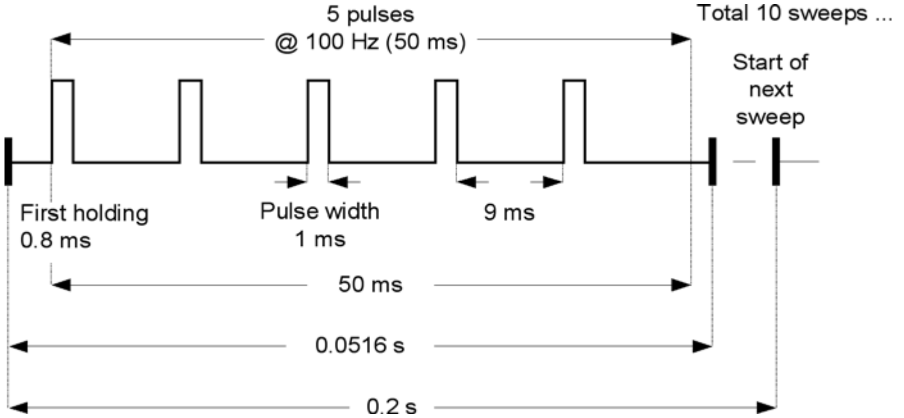


Figure 4.7: Default "Theta" conditioning stimulus.

- Number of sweeps: 10
- Sweep start-to-start: 0.2 s (5 Hz)
- Waveform length: 50 ms
- Sweep length: 0.0516 s
- Trains per sweep: 1
- Pulse level: 20 mV
- Delay to first train: 0 ms
- Train duration: 50 ms
- Intertrain interval: not applicable
- Pulses per train: 5
- Train frequency: 100 Hz
- Pulse width: 1 ms
- Sampling interval: 100 µs (10 kHz)

LTD: A single pulse at 1 Hz is delivered for 15 minutes.

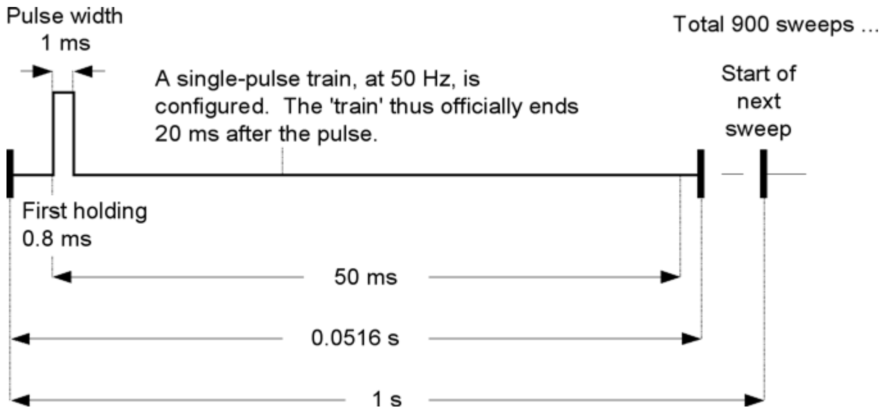

Pulse width
1 ms
Total 900 sweeps ...
A single-pulse train, at 50 Hz, is configured. The 'train' thus officially ends 20 ms after the pulse.
Start of next sweep
First holding
0.8 ms
50 ms
0.0516 s
1 s

Figure 4.8: Default “LTD” conditioning stimulus.

- Number of sweeps: 900
- Sweep start-to-start: 1.0 s (1 Hz)
- Waveform length: 50 ms
- Sweep length: 0.0516 s
- Trains per sweep: 1
- Pulse level: 20 mV
- Delay to first train: 0 ms
- Train duration: 20 ms, but not relevant
- Intertrain interval: not applicable
- Pulses per train: 1
- Train frequency: 500 Hz, but not relevant
- Pulse width: 1 ms
- Sampling interval: 100 µs (10 kHz)

Postsynaptic Pairing

In postsynaptic pairing a stimulus is delivered to a postsynaptic cell in association with the presynaptic conditioning stimulus. This option therefore requires that you have a postsynaptic command channel enabled, on the Inputs/Outputs tab.

Two types of postsynaptic pairing can be configured. Usually used in voltage-clamp experiments only, you can set a voltage step to be held for the duration of the conditioning stage. The postsynaptic command shifts to the set voltage after the first holding in the first sweep of the conditioning protocol and maintains this until the last holding in the final sweep, including intersweep periods where there is otherwise no output.

More often used in current-clamp experiments, you can also opt to deliver a single postsynaptic pulse, following a stipulated time after the end of the presynaptic stimulus.

For the delayed pulse option, in order that the postsynaptic waveform is timed with respect to the end of the presynaptic stimulus, you must use a digital channel for presynaptic stimulation, and the analog postsynaptic channel of the same number for the paired stimulation, e.g. digital OUT #0 for presynaptic stimulation, and analog OUT #0 for the postsynaptic command. It is possible to pair postsynaptic stimulation with an analog presynaptic command, by enabling, say, analog OUT channels #0 and #1 for presynaptic and postsynaptic commands respectively, but in this case you need to time the postsynaptic waveform from the start of the sweep. The delay to the postsynaptic command, in this case, is not automatically calculated from the end of the presynaptic stimulus.

Membrane Test

The LTP Assistant no longer allows you to set up a resistance test to monitor seal and access resistance on a postsynaptic intracellular recording electrode during baseline stages.

In order to monitor access resistance over the length of an experiment, the experiment will need to be manually setup using the protocol editor with the stimulus tab's Membrane Test Between Sweeps, and sequencing keys.

Here it is recommended that you configure some settling time following the end of the presynaptic stimulus, before the Membrane Test test pulse is generated. This allows the cell some time to settle following stimulation from the presynaptic cell.

Statistics

The LTP Assistant allows three online statistics measurements to be recorded for all input signals during baseline stages. These are enabled, in the first case, by checking the statistics checkbox for the channels you want, on the Baseline tab. Then later, when you have closed the LTP Assistant and started the experiment, you must set appropriate search regions in the Scope window, to capture the measurements properly. The three measurements are:

- Peak amplitude
- Peak time
- Slope

Two different statistics search regions and a baseline region are used to measure these. Search region 1 is used to measure peak amplitude and peak time, so once you are acquiring stable baseline data you should drag the “1” cursor pair in the Scope window to capture the entire event evoked by the stimulus. Search region 2 is configured to measure slope—by linear regression for all the data points within the region—and is intended to measure the rising slope of the event. Cursor pair “2” should then be positioned to capture this feature of the event. The baseline should be set to a section of the sweep where there is no activity (see figure below):

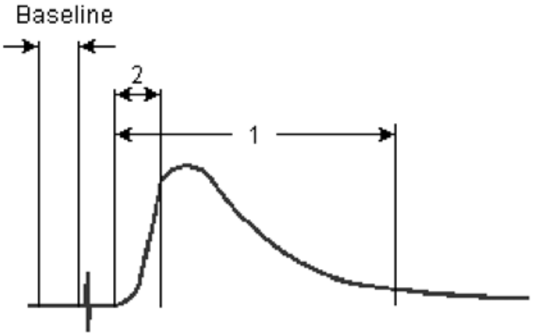


Figure 4.9: Intended statistics baseline and search region positions.

The regions you set are saved automatically when the protocol closes. This means that a baseline protocol opening automatically after a conditioning stage retains the settings you made in the earlier baseline stage.

Statistics measurements are displayed continuously, for the length of the experiment, in the Online Statistics window, from where they can be saved as an *.sta file in the normal fashion. Note that under default settings, the Online Statistics window is saved at the end of each protocol run. This means that so long as the final baseline protocol runs to completion, the statistics for the entire experiment are saved (but with gaps where, for example, a conditioning protocol with no statistics recorded was run). If, however, you have set a very long baseline protocol and stop the experiment manually, statistics are not automatically saved. In this case you need to save them manually, with File > Save As.

As with all other protocol features in the LTP Assistant, the full set of statistics configuration options can be accessed from the protocol editor, if you want to take more or different measurements from the data.

5. Clampex Tutorials

This chapter presents information to help you actually perform experiments. We walk you through a step-by-step tutorial on setting up experiments. We discuss how Clampex can be used to further manipulate data files that have already been recorded. We discuss how to use the Quick Graph feature in Clampfit to automatically plot an I-V (current vs. voltage) curve while Clampex acquires data. We finish by presenting several scenarios for doing different types of experiments.

I-V TUTORIAL

This tutorial is designed to walk a user through the steps of setting up various types of experiments in Clampex. The goal of this section is to teach a user how to set up Clampex to perform a given experiment, rather than discuss each feature of Clampex in detail. For more information on a given feature, a user can access the online Help by pressing <F1>. You may also want to first follow the tutorial Setting Up Clampex for Data Acquisition, installed on the desktop, which covers the basic setup procedures for a Digidata digitizer and the Axopatch 200B and MultiClamp 700B amplifiers.

Goal of the experiment: To examine the I-V relationship of whole-cell calcium currents from cultured neurons in response to the application of a drug.

The experiment consists of:

1. Obtaining a whole-cell recording.
2. Monitoring access resistance.
3. Performing an I-V test.
4. Monitoring the peak current at a single voltage.
5. Applying a drug while measuring the peak current, and then performing an I-V test.
6. Washing off a drug while measuring the peak current, and then performing an I-V test.
7. Monitoring access resistance at end of experiment.

Hardware used:

- Axopatch 200B amplifier, headstage set to β = 0.1.
- Electronically controlled solution changer.
- Digidata 1440A digitizer.

Select Digitizer

One of the first steps in performing an actual experiment is to select the appropriate hardware for acquisition. This is done with the Configure > Digitizer menu item. By default, Clampex is installed with the "demo" digitizer active. We will continue to use the demo digitizer for this example, but you must select and configure the digitizer before doing an actual experiment.

Creating Input and Output Signals in the Lab Bench

The Lab Bench is used to set the scaling and offset factors for all the input and output channels used in an experiment.

In the example above, an Axopatch 200B amplifier is being used in voltage-clamp mode to record membrane currents in response to square voltage steps. We will set up Clampex to acquire the membrane currents on Analog IN #0, and to output the voltage command on Analog OUT #0.

Setting Up Input Signals

1 To set up the Lab Bench for this example, select Configure > Lab Bench.

2 Select the Input Signals tab, and then highlight the Digitizer channel Analog IN #0. You will notice that there are already a number of default signals associated with Analog IN #0.

3 Add another signal to Digitizer channel Analog IN #0, by pressing the Add button.

4 Give the signal a unique name such as "Im".

5 Now set the scaling for this new signal. This is done in two steps: by entering the units for the signal and by setting the scale factor.

6 To set the signal units, in the Scaling section select the unit prefix "p", and enter the unit type as "A".

7 To set the scale factor, we will use the Scale Factor Assistant. All we need to know is the gain settings of our amplifier, which can simply be read off of the front of the instrument. In our example, the signal is membrane current, and we will assume the Axopatch 200B amplifier is set to an α value of 1 and a β value of 0.1.

8 Press the Scale Factor Assistant button and indicate that you have an Axopatch 200 series amplifier connected to Analog IN #0.

9 Press the Next button.

10 Since the experiment will be performed in voltage-clamp mode, select the V-Clamp Mode Setting.

11 Set Config Setting to Whole Cell (β = 0.1) to match the β setting of our Axopatch 200B amplifier.

12 Set Preferred Signal Units to "pA".

13 Set Gain to 1 to match the α setting of our Axopatch 200B amplifier.

14 Press Finish.

The scale factor of 1e^{-4} V/pA was automatically computed and entered as the Scale Factor. For most amplifiers, there is no need to adjust the offset and therefore we will leave the offset value at its default of 0.

For this example, set the gain (α) to a value of 1. This is referred to as "unity gain". The setting of unity gain is very important when you enable telegraphs, a functionality that we discuss in the next section. For now, be sure to set the gain to 1, and set up the Lab Bench to reflect this. We will discuss how Clampex deals with changing the gain in the telegraph section.

Clampex has an online software RC filter. We can specify both a highpass and lowpass filter to be applied to this Analog IN channel. For our example, leave this turned off.

We may want to have an additional gain applied to our signal before it is digitized, to ensure that we are maximizing the dynamic range of our digitizer. Since the Axopatch 200B amplifier has a number of different output gains, it is unlikely that you will need to use any additional hardware conditioning for the membrane current signal. You have now finished setting up the Analog IN channel to record "Im" signals from the Axopatch 200 amplifier.

We will now configure Analog IN #1 to record the membrane potential. You will add the 10 V_m OUTPUT signal from the Axopatch 200 amplifier to Analog IN #1, and set the scaling factor to record voltage in mV. The Scale Factor Assistant should not be used for this signal, as it assumes that the Analog IN channel is connected to the scaled output of the Axopatch 200 amplifier, and not to other signals such as 10 V_m. Since the gain of the 10 V_m OUTPUT of the Axopatch 200 amplifier is x 10, manually set the scale factor (V/mV) to "0.01" for the signal "V_m".

We need to consider whether, with a gain of 10, the signal that we are recording adequately utilizes the dynamic range of the digitizer. Typically, the cellular membrane potential will be in the range of ±100 mV, and with a gain of 10, the Axopatch 200B amplifier will output a voltage in the range of ±1 V (100 mV x 10). Given that the dynamic range of the digitizer will be ±10 V it is apparent that only ~10% of the dynamic range will be utilized (±1 V/±10 V). But, for a 16-bit digitizer such as the Digidata 1440A digitizer, while not optimal, this is resolved into about 6,500 steps, which is still quite adequate.

Setting Up Output Signals

Setting up the output signals is very similar to setting up the input signals. You need to determine an output scaling factor and offset for each signal that you create on a given Analog OUT channel. You will notice that there are three Signals already associated with Analog OUT #0 by default. We can add another signal, "VmCmd" to Analog OUT #0. For this new signal, set the Signal units to "mV" and use the Scale Factor Assistant to set up the scale factor. In our example, we would indicate that the "VmCmd" signal is connected to an Axopatch 200 amplifier, with a Mode Setting of

“V-Clamp”, and that the Ext. Command Input of 20 mV/V on the Axopatch 200 amplifier is connected to Analog OUT#0.

Setting the Holding Level

The holding level can be set in a number of locations in Clampex, including the Lab Bench if you check the appropriate Configure > Overrides checkbox. Default behavior has holding levels for each output channel set by the currently loaded protocol (from the Outputs tab in the Protocol Editor). By taking the Lab Bench Overrides option, one holding level is maintained on each channel irrespective of the protocol that is loaded. However, wherever the override is set for, immediate changes to the holding level can be made from the Real Time Controls. In addition, if you are running the Membrane Test, you can change the holding level from within its window. The Membrane Test returns the holding level to where it was at the time the test was opened.

Telegraphs

A convenient feature of Clampex is the ability to detect various amplifier settings through “telegraphs”. These telegraphs allow you to change the settings of various amplifier controls, such as the gain, without having to reconfigure the Analog IN channels after each change in amplifier gain. For example, on an Axopatch 200 amplifier, the state of the gain (α), filter frequency and capacitance compensation settings can be directly used by Clampex. You may recall that we set the gain of our Analog IN #0 signal “Im” to a value of 1, or unity gain. When using telegraphs, Clampex performs all calculations relative to this unity gain, and therefore it is very important that unity gain is set up correctly for the signal that will be telegraphed. The telegraphs are configured through the Configure > Telegraphed Instrument menu item.

In our example, we will indicate that an Axopatch 200 CV 201 is being used, and that Analog IN #0 is connected to the scaled output of the amplifier. Since we want to telegraph all three settings, we simply connect the appropriate outputs of the Axopatch 200 amplifier to the appropriate telegraph ports on the Digidata 1440A digitizer. Now, if we close the Configure > Telegraphed Instrument dialog box and return to the Lab Bench, we notice that the “Telegraphs” section at the bottom of the inputs tab reports the current status of these three telegraphed parameters. If you enable the telegraphs, you notice that the gain setting in the Scale Factor Assistant is automatically inserted via the telegraphs. Furthermore, any changes that are made to these parameters are automatically updated and used by Clampex.

Using Membrane Test

The first step in establishing a whole-cell recording is achieving a gigohm seal.

Before starting Membrane Test, you need to ensure that it is configured correctly using Configure > Membrane Test Setup. By default, Membrane Test is set up to record input signals from the channel Analog IN #0 using a signal setting of I_MTest 0 (pA), and to output Membrane Test signals to the channel Analog OUT #0 on a signal setting of Cmd 0 (mV). We will change the input signal to “Im”, and the Membrane Test output signal to "VmCmd" to match the signals that were just created in the Lab Bench. You can also adjust the amplitude of the pulse when Membrane Test is started and set initial holding levels. We will set the initial holding level to a value of 0 mV.

After configuring Membrane Test, it can be opened from Tools > Membrane Test, or from the toolbutton on the Acquisition toolbar. After starting Membrane Test in Bath mode, you will observe the current response to square voltage steps.

You will see that Clampex continuously reports the Seal Resistance value. This value can be logged to the Lab Book at any time by pressing the => Lab Book button.

You can also adjust how often the pulses are delivered using the frequency slider.

In some cases, you may want to hyperpolarize the cell to aid in seal formation. This is done by setting the "Holding (mV)" value.

Editing a Protocol

The details of an acquisition are defined in the acquisition protocols. As this is a fairly complex issue, we discuss only the features that are relevant for setting up our example. In this case, we describe how to build two protocols: one that performs an I-V, and one that repetitively steps the cell to the same voltage. We have provided these demonstration protocol files for you, named tutor_1a.pro and tutor_2a.pro, as well as the sequencing key file, named tutorA.sks, in ..\Program Files\Molecular Devices\pCLAMP10.0\Sample Params.

Protocol 1: Conventional I-V

To get a fresh protocol, select Acquire > New Protocol, which generates and opens a new protocol. We will examine each relevant tab of the protocol to ensure that the protocol is set up correctly.

Mode

Select Episodic stimulation as the Acquisition Mode. Since our I-V will step from -80 to +60 in 10 mV increments, we need to have 15 steps. Therefore, we set the sweeps/run to a value of 15. We will keep the Start-to-Start time for sweeps set to "Minimum" to have the protocol execute as fast as possible. Since we will sample the data at 10 kHz, the sweep duration should be set to 0.2064 s, to ensure that each sweep is long enough to gather enough data.

Input and Output Channels

In the Inputs tab, you activate the Analog IN channels that you are acquiring from, and indicate which signal that you created in the Lab Bench you would like to record on each channel. In our example, you would activate Channel #0 and Channel #1, and specify the "Im" and "Vm" signals respectively.

In the Outputs tab, you define which Analog OUT channels and signals are active. Set the Channel #0 to the signal "VmCmd" that you created in the Lab Bench. We will also define a holding level of -80 mV.

Waveform

The next logical step is to define the stimulus waveform on the Waveform tab. We set the source of the waveform to be defined by "Epochs", which are described by the table on the form. For this protocol, we disable digital outputs. You can change the intersweep holding level, but currently we will leave it at the default value of "Use Holding". The only item remaining to be done is the description of the actual epochs, which is performed in the Epoch Description table. We will define the A, B, and C epochs as step epochs by selecting "Step" as the Type. You can also use the <PgUp> and <PgDn> buttons to cycle through the epoch types. For our example, we need to make the following Epoch descriptions:

Table 5.1: Epoch descriptions for Protocol 1.

	Epoch A	Epoch B	Epoch C
Type	Step	Step	Step
Sample rate	Fast	Fast	Fast
First level (mV)	-80	-80	-80
Delta level (mV)	0	10	0
First duration (ms)	50	100	50
Delta duration (ms)	0	0	0
Digital bit pattern (#3-0)	1111	0000	0000
Digital bit pattern (#7-4)	0000	0000	0000
Train rate (Hz)	0	0	0
Pulse width (ms)	0	0	0

When finished, you can preview the waveform by pressing the Update Preview button at the bottom of the form. In fact, it is often very useful to have the Waveform Preview window open while you are creating the protocol, and simply press the Update Preview button every time you want to see the effects of something you have changed. It is very handy to shrink the size of the Waveform Preview and keep it displayed in a corner of the screen.

Trigger

We will leave the Trigger tab in its default state for this protocol.

Stimulus

The Stimulus tab is where you define additional stimulation parameters, such as pre-sweep trains, leak subtraction and arbitrary values from a user list. For this demonstration, using the demo digitizer, we have not added P/N leak subtraction to our protocol in the Stimulus tab, however, when recording from real cells you might wish to add P/N leak subtraction to the protocol. We have defined a P/N leak subtraction stimulus in the protocol, but left it disabled. Check the checkbox to view our configuration: 5 subsweeps of polarity opposite to the waveform to occur before each individual sweep, with a minimum start-to-start time. The signal to which leak subtraction is applied is I_m. You can optionally display the raw or corrected data.

Statistics

Clampex can display online various spike-oriented shape statistics on an acquired signal. Go to the Statistics tab of the protocol editor and check the box Shape Statistics. In our example, we will measure the peak amplitude and time of peak on the "I_m" signal.

If we were recording actual inward calcium currents, we would want to set the polarity of the peak detection to negative. Since we are using the demo driver, we will set the polarity to positive to illustrate the functionality of the statistics display. We can define a search region and a baseline region either in terms of time or epochs. We'll define the search region as Wave 0 Epoch B, and the baseline region as Wave 0 Epoch A. If you want to ensure that you are searching the correct regions of the waveform, use the Update Preview button. If Shape Statistics are enabled, the Waveform Preview window shows the Baseline and first Search region used in the peak detection.

The statistics can be automatically saved whenever an acquisition is saved to disk, although if this option is not enabled, you can always manually save the statistics from the Online Statistics window with File > Save As. For now, we will not enable this option.

That completes setting up your first protocol. Save the protocol using the Acquire > Save Protocol As menu item, with a name such as "tutor_1.pro".

Protocol 2: Repetitive Current Stimulation

The second protocol is one that is designed to repetitively activate the current by giving step pulses from -80 to +10 mV. This type of protocol if useful when one wants to monitor the effect of a drug over time.

To create this protocol, we will modify the existing protocol that we just created. Use the Acquire > Edit Protocol menu item to edit the current protocol. In the Mode/Rate tab, adjust the Sweeps/run to 6. For demonstration purposes we have made the protocol as fast as possible, but for real use you might want to adjust the Sweep Start-to-Start Interval to 10 s and re-save the protocol. In the Waveform Channel #0 tab, change the First Level of Epoch B to +10 mV and the Delta level to 0. This creates a protocol that steps to +10 mV for 100 ms in every sweep. Press the Update Preview button to ensure that you have changed the waveform correctly. Save the protocol under a new name, such as "tutor_2.pro" using Acquire > Save Protocol As.

At this point, you have configured Clampex correctly for the hardware that you are using for the experiment, and have defined the two protocols that you need to use in order to perform the experiment. Additionally, you have learned about the Membrane Test, which aids in the formation and monitoring of a gigohm seal.

Before you save any data to disk, define the file names and locations where the data will be saved (and opened from). This is done in the File > Set Data File Names dialog box. For this example, we will use "tutor" as the filename prefix. Select long filenames, disable the date prefix, and type "tutor" in the file name field.

Using Sequencing Keys

Sequencing keys are an extremely powerful and flexible feature of Clampex that allow you to define an entire experiment composed of several different protocols, interspersed with Membrane Tests, and parameter changes such as digital outputs to control solution changes. Sequencing keys allow you to associate an event such as running a protocol, running a Membrane Test, or changing a parameter, with an individual keystroke and its associated toolbutton. Each keystroke can then be linked to another keystroke, allowing the user to set up a chain of events that define an entire experiment.

In this section we discuss how to use the information from the previous sections to design a complete experiment that can be run by the touch of a single button. Before setting up the sequencing keys, it is often helpful to determine the flow of the experiment. In our example, the experiment is to examine the I-V relationship of whole-cell calcium currents from cultured neurons in response to application of a drug.

We will use the sequencing keys to automate the experiment after whole-cell access has been achieved. We can think of the experiment in terms of the protocols and events, such as solution changes, that we use.

Table 5.2: Experimental steps, events, and protocols.

Step	Event	Protocol
1	Obtaining a whole-cell recording	

Step	Event	Protocol
2	Monitoring access resistance	Membrane Test
3	Performing an I-V	Tutor_1.pro
4	Monitoring the peak current at a single voltage	Tutor_2.pro
5	Start applying a drug	Digital Out ON
6	Monitoring the peak current at a single voltage	Tutor_2.pro
7	Monitoring access resistance	Membrane Test
8	Performing an I-V	Tutor_1.pro
9	Washing off the drug	Digital Out OFF
10	Monitoring the peak current at a single voltage	Tutor_2.pro
11	Monitoring access resistance	Membrane Test
12	Performing an I-V	Tutor_1.pro

Now that we have established what the events are that we would like to define in the sequencing key setup, we can proceed to configure a sequencing series.

You set up the sequencing keys through the Configure > Sequencing Keys menu command or, if the Sequencing Toolbar is active, by pressing the first button on the Sequencing Keys toolbar:

- Once you open the Sequencing Keys dialog box, select New Set to define a new set of sequencing keys.
- To define a new event in the sequence, press the Add button. You are prompted as to what key you would like to define. Choose the default of <Ctrl+1>.
- You can define one of four operations that can occur when the key is activated: analog and digital holding levels, Membrane Test, protocol and message prompts.
- For each key you define, use the Sequencing tab to define what happens when the operation completes. In this case, we link each event to the next key, so that we can set up a continuous sequence of events.
- We can also define when the next event will occur. For example, we can specify that we would like the next event to occur when the current acquisition finishes, or after some fixed time has expired.

In our example, we will define <Ctrl+1> to run the Membrane Test for 1 s, and when it completes, we will immediately run the Tutor_1 protocol while saving the data to disk.

1 In the Operations tab, select Membrane Test.

2 On the Sequencing tab define the Next key as <Ctrl+2>.

3 Specify that you want to "Start next key" after 1 s has elapsed from the start of the key.

4 Press OK to finish defining <Ctrl+1>.

5 Now add the <Ctrl+2> key by clicking the Add button.

6 Set the Operation to "Protocol".

7 Specify the Action as "Record".

8 Next, use the Browse button to browse to the protocol file Tutor_1.

9 Run the protocol only once by ensuring the repetition count is 1.

10 On the Sequencing tab, set the sequence to link to the next key of <Ctrl+3> when the acquisition finishes.

11 Add the <Ctrl+3> sequencing key and this time link the key to the protocol Tutor_2 using the same procedure as above, and choose <Ctrl+4> as the Next key in the Sequencing tab.

To toggle an electronic solution changer on and off, you can use the Digital OUT Bit Pattern on the Operations tab. Simply specify which digital bit you would like to turn on or off. For example, you could connect the solution changer to Digital OUT #0 and you would enable Digital OUT #0 when you wanted to activate a solution change, or disable the Digital OUT when you want to end the solution change.

In our experiment, the <Ctrl+4> key is defined to flip Digital OUT #0 high:

1 Add the <Ctrl+4> sequencing key.

2 In the Operations tab, choose Parameters and enable the Digital OUT Bit Pattern.

3 Uncheck all bits except bit 0 (note that you may have one bit unavailable here, overridden by the "high during acquisition" checkbox in the Lab Bench).

4 Choose <Ctrl+5> as the Next key in the Sequencing tab.

In our example, we want to keep our Digital OUT high until we set it back to low after we observe an effect of the drug on the cell. To keep the digital bits high when we run subsequent protocols, we must set Use digital holding pattern from Lab Bench in Configure > Overrides. When this option is enabled, any change to the digital outs are stored in the Lab Bench, and are only changed when the Lab Bench is changed, and not by loading and running a protocol.

Using the outline of the experiment you can set up the entire sequence. Try using the Copy button to create new keys, followed by Properties to edit the new key. When you are done the sequence should look like:

Table 5.3: Key sequence for experiment.

Key	Next Key	Type	Description
<Ctrl+1>	<Ctrl+2>	Membrane Test	Start-to-Start = 1
<Ctrl+2>	<Ctrl+3>	Protocol	Record using “Tutor_1.pro”. 1 reps. Protocol time takes priority
<Ctrl+3>	<Ctrl+4>	Protocol	Record using “Tutor_2.pro”. 1 reps. Protocol time takes priority.
<Ctrl+4>	<Ctrl+5>	Parameters	Digital = 00000001. Start-to-Start = 1
<Ctrl+5>	<Ctrl+6>	Protocol	Record using “Tutor_2.pro”. 1 reps. Protocol time takes priority.
<Ctrl+6>	<Ctrl+7>	Membrane Test	Start-to-Start = 1
<Ctrl+7>	<Ctrl+8>	Protocol	Record using “Tutor_1.pro”. 1 reps. Protocol time takes priority
<Ctrl+8>	<Ctrl+9>	Parameters	Digital = 00000000. Start-to-Start = 1
<Ctrl+9>	<Ctrl+Shift+1>	Protocol	Record using “Tutor_2.pro”. 1 reps. Protocol time takes priority
<Ctrl+Shift+1>	<Ctrl+Shift+2>	Membrane Test	Start-to-Start = 1
<Ctrl+Shift+2>	None	Protocol	Record using “Tutor_1.pro”. 1 reps

You can save the sequence with the Save Set button, naming it something like “Tutorial.sks”.

After closing the Sequencing form, you can start the entire experiment by pressing <Ctrl+1>, or the second key in the Sequencing toolbar.

A text description of a sequencing set can be copied to the clipboard with the => Clipboard button, and then pasted into a word processor, to allow you to print out the sequencing series for your lab notebook.

Displaying and Measuring Acquired Data

After you have acquired data, Clampex provides some basic analysis functionality. It should be noted that data is best analyzed in Clampfit, but Clampex allows you to browse data and perform simple measurements.

You can open data files by selecting File > Open Data. This dialog also provides a number of options that control how the data is displayed once it is read into Clampex. For example, you can determine if each data file is opened into a new window, how to control the axis scaling, and you can set the number of sweeps or signals that you would like displayed. Open the last data file that was recorded by selecting File > Last.

Once the data is opened in the Analysis window, there are four cursors that can be moved by clicking and dragging. Many of the options that control the Analysis windows are available by clicking the right-mouse button while in various areas of the window.

For now, we will simply measure the peaks of our calcium currents that we acquired during the last I-V of the sequencing series, which was the last file that was recorded, and which should now be displayed. We will measure the minimum value between cursors 1 and 2:

→ Set cursors 1 and 2 so that they delimit the region of the peak inward calcium current.

→ Select Tools > Cursors > Write Cursors. This function can also be performed by pressing the “check” toolbutton at the top left of the Analysis window or from the Analysis toolbar.

→ Double-click on the Results window at the bottom of the screen, or select Results from the Window menu. Observe that for each trace, a number of parameters were measured.

→ You can control which measurements are made, and in what order they are displayed, by selecting View > Window Properties when the Results window is displayed.

→ You can save these measurements to a text file by selecting File > Save As.

There is extensive control over various features of the Analysis window. The online Help covers this in detail. A useful reminder is that most properties can be set by right-clicking in various areas of the Analysis window. You can set various properties of the cursors and the Analysis window through this functionality.

Real-Time I-V Plotting in Clampfit

Despite the power of Clampex itself, we will now coordinate its acquisition with I-V plotting in Clampfit. For this demonstration you may first wish to return Clampfit to its factory default values. Do this by closing all Axon Instruments software and then running the program Reset to Program Defaults found in the Axon Laboratory program folder. Choose Clampfit as the application to return to default values.

Now load both Clampfit and Clampex and in Clampex load your protocol file tutor_1 or the protocol file provided as tutor_1a.pro. In Clampfit, check “Apply automatic analysis” in Configure > Automatic Analysis. Using the Record function in Clampex (do not use the sequencing keys yet) record a file, which should then appear in an Analysis window in Clampfit.

We will now use this file to create our I-V plotting routine and then we will set up Clampfit to run an Automatic Quick I-V Graph concurrently with our experiment in Clampex:

1 With the focus on the Analysis window in Clampfit, select Analyze > Quick Graph > I-V. We want to plot the peak current I_m against the command voltage VmCmd.

2 For the X Axis Waveform group choose Epoch, and "Epoch B level" for the X axis of the plot.

3 For the Y Axis, for Signal choose "Im". We want to detect the positive peak in Epoch B so select Peak and choose "Positive" for the Polarity. For Region choose "Epoch B - Waveform 0".

4 We apply a smoothing factor of 5 points in peak detection.

5 Under Destination Option choose "Append" because we want to see all I-V curves in the experiment plotted together.

6 When you close the form a graph window will appear with an I-V plot.

7 Minimize all applications on the Windows desktop except Clampex and Clampfit. Move the mouse to a blank part of the Windows taskbar, right-click and choose Tile Windows Horizontally to give equal place to the two applications. You may now want to arrange the Windows within each application to make maximal use of the desktop space. In Clampfit minimize or close all windows except the Analysis window and the Graph Window and arrange the windows with a Window > Tile Vertical command in Clampfit.

8 Before beginning the experiment, open Configure > Automatic Analysis, check Generate Quick Graph and select I-V. In this demonstration, make sure that Prompt for Arithmetic is not checked. When you run an actual experiment at slower rates of acquisition, you would enable Subtract Control File in order to subtract control sweeps from the sweeps to be analyzed automatically and plotted as I-V or Trace versus Trace graphs.

9 In Clampex, load the tutorial.sks sequencing file or the sequencing file provided as tutorA.sks with the command Configure > Sequencing Keys > Open Set. If the Online Statistics window is visible and has data in it, clear the data with the command Edit > Clear Statistics. Now start the experiment by pressing the <Ctrl+1> toolbutton. As the data is acquired you should see six I-V curves plotted in the Graph Window.

MEMBRANE TEST TUTORIAL

This tutorial is designed to provide a quick overview of the performance and special features of the Membrane Test. It takes 10–15 minutes to complete the tutorial. It is important that you proceed through the tutorial in sequence for this tutorial to be useful.

1 Set Clampex to demo mode

Go to the Configure > Digitizer dialog, select the Change button and select "Demo" from the Digitizer Type list.

2 Open the Membrane Test

Push the circuit diagram toolbar button or select Tools > Membrane Test. Confirm that the OUTPUT signal is Cmd 0 (mV) and that the INPUT signal is I_Mtest 0 (pA). If this is not the case, change the settings in Configure > Membrane Test Setup (you can do this without closing the Membrane Test window).

3 Select the Membrane Test mode

Click on the Cell button, and make sure that input channel #0 is selected. Then, click on the Play button to start the membrane test.

4 Change the voltage pulse amplitude

Change the pulse amplitude in the Pulse field to 20 mV, and then from 20 mV to 10 mV either by clicking on the down arrow button or by typing in the number (followed by <Enter> on the keyboard). Notice that the capacitive transients from the current record in the Scope display decrease in amplitude. Now change the Pulse amplitude back to 20 mV and observe that the capacitive currents are larger.

5 Change the scale in the scope display

Click the up arrow button (on the left side of the window by the Y axis) a couple of times to change the amplitude scaling. Push the Auto Scale button next to the meters and observe the change.

6 Change the measurement update rate

Move the slider marked Slower – Faster. The frequency is shown to the underneath. Notice near the Slower end the calculations fail because there are too few points in the decay of the transient. As you move the slider, observe that the frequency value (measurement update rate) changes. The capacitive transient current records change, as well as the time scale on the X axis. The membrane parameter values also change, depending on the accuracy of the calculation, which depends on the accuracy of the curve fit (red line on the decay phase).

7 Check the Averaging checkbox to enable averaging

Click the Averaging checkbox to enable the averaging function. Notice that the noise of the capacitance signal decreases in the Online Statistics window. Click on the Averaging checkbox again and observe the noise increase (wait a moment for the new data to appear). Note also that the recording speed (displayed below the Slower/Faster slider) increases dramatically.

8 Change the number of pulses in the pulse train

Open Configure > Membrane Test Setup. In the Pulse Train Protocol section, set Number of pulses in train to 4. Change the step level (mV) to 40. Select OK to return to the Membrane Test Window. Click on the Pulse Train button and observe the 4 pulses appear sequentially in the Scope display. Notice also that the values in the Online Statistics window do not update during the pulses.

9 Save values to the Lab Book

Push the => Lab Book button. The values of the five Membrane Test parameters are saved to the Lab Book. Confirm this by opening the System Lab Book window. Return to the Membrane Test window.

10 Save, then view results

Make the Online Statistics window active by clicking on its title bar. Select File > Save As to save the data in the Statistics window to a *.sta file. Note the file name and destination folder and click OK. Close the Membrane Test window. Open the file that was just created by selecting File > Open Other > Results. Change the Files of type field to "All Files (*.*)". Select the *.sta file that you just saved. It is also a good exercise to import the file into a spreadsheet program such as Excel, SigmaPlot, or Origin.

Conclusion

This concludes the Membrane Test Tutorial. Be sure to reconfigure the Digitizer for data acquisition. If changes were made for the tutorial, you may need to change OUTPUT or INPUT signals to their pre-tutorial settings in the Membrane Test Setup dialog.

SCENARIOS

Cell-Attached Patch-Clamp Single-Channel Recording

Clampex's Membrane Test tool can be used to monitor seal resistance until a suitably high-resistance seal is obtained. Currents can then be recorded in Gap-free mode, or alternatively in Variable-length events mode. Important information like the time of the recording, the holding potential and telegraph information are stored in the file header and can be retrieved by selecting File > Properties. Additional information about the experimental conditions can be added as a voice tag or a typed comment tag. The toolbar buttons can be customized to present your preferred form of comment tag.

The simplest procedure would be to use gap-free recording. If event frequency is reasonably high, gap-free recording is probably the best mode to use. However, the event frequency is often not predictable until the data starts to appear. Based on the nature of the patch, it may be less than a second, or more than a minute, before enough events occur to permit a good analysis of the events. One way to be prepared is by having two protocols ready to run: a gap-free protocol if event frequency is high, and an event-triggered protocol if event frequency is low.

Sequencing keys can then be used to automatically load one protocol or the other, based on your initial impressions of the data. At this point, your subjective experience of your preparation and your specific experimental goals will determine how to proceed. You need to consider how long the patch is likely to be stable, and how many events are sufficient to make a particular experimental measurement.

For example, if you just want to generate amplitude histograms at multiple voltages to calculate a single channel I-V, then you probably just need a few hundred events at each voltage. As a consequence, the recordings may be relatively short. On the other hand, if you are primarily interested in knowing about channel kinetics, then relatively long recordings may be in order. Some experience with how long the patch is likely to be stable is another important factor, and also whether the single-channel activity itself is going to be stable or show rundown. The real-time display of threshold-based statistics may be useful for making decisions about how to best get the desired data from the patch. For example, you should be able to detect rundown by generating a running estimate of event frequency in the statistics display.

Since the Analog OUT channels and Digital OUT channels allow the user to change experimental parameters in real time, it is possible to get a clear representation of their cause and effect relationships in experiments. By using a BNC "T" connector to send the same signal to a peripheral device such as a valve controller, and to one of the acquisition channels, you can simultaneously record the output signals in addition to the experimental response (recordings may then be viewed in a continuous or segmented format). However, this dedicates an entire Analog IN channel to a parameter that changes only occasionally. For many applications, the better approach is to take advantage of the fact that changes in the output signal levels and the digital outputs can be tagged internally. Tags are added to the data file each time you change an output voltage or digital output. Or, within the Trigger tab of the Protocol Editor, you can select the External Tag option, and use the TRIGGER IN/TAG BNC on the digitizer to mark changes in the file while recording.

If you want to change the holding potential to determine the voltage dependence of conductance or kinetic parameters, the simplest thing is to stop the trial, adjust the V_{out} channel or the amplifier setting, and start another gap-free trial. However if you want to see if there are immediate (i.e. non-stationary) effects of the voltage jump, then you can change the V_{out} signal in real time and have a comment automatically inserted into the recording at the time of the voltage jump. Data ranges can be selected with the cursors and saved as separate binary integer files to be analyzed independently later.

When event frequency is low in single-channel experiments, a great deal of disk space can be conserved if the system only records during the time when the channel is active. Variable-length events mode is ideally suited for this. The pretrigger interval can be set to record data before the trigger, to ensure that baseline drift can be evaluated and corrected in the subsequent analysis of the data. The pretrigger interval also sets the amount of data that is saved after the signal recrosses the threshold, so that with this value set correctly, events are saved in their entirety.

When recorded at a high bandwidth, single-channel data may be too noisy for event-triggered acquisition to be practical. Baseline drift can also be a problem with threshold-based event detection. Event-triggered acquisition can also require that the baseline is adjusted in real time to keep the position of the data constant relative to the trigger threshold. One option to get around these problems is to acquire the data on two Analog IN channels, one filtered and the other set at a higher bandwidth. Trigger information can then be taken from the filtered signal. The higher bandwidth setting on the second Analog

IN channel preserves single-channel data for analysis. Since the filter settings in the Clampex Lab Bench are applied to each channel separately, this can be accomplished entirely within Clampex. For example, the Clampex lowpass filter of Analog IN channel 0 could be set to 5 kHz, while the lowpass filter of Analog IN channel 1 is set to 1 kHz along with a highpass filter set to 0.5 Hz. This basically makes Analog IN channel 1 AC-coupled and essentially removes baseline drift. With this setup, events are detected on the AC-coupled 1 kHz channel, while single-channel data is saved at 5 kHz.

An important caveat to this approach is that the minimum event duration is in part a function of the filter setting on the trigger channel, and in part a function of the filter setting on the data channel. Single, brief events may not trigger acquisition. However, brief events within bursts of activity will be resolved at the higher bandwidth. Therefore, this approach would best be applied to the analysis of kinetics within bursts of activity. If you are changing the holding potential while conducting a variable-length event-driven acquisition, you should also keep in mind that the change in channel amplitude also alters the event-detection sensitivity.

Isolated Patch-Clamp Single-Channel Recording

The big advantages of isolated patch-clamp recording (whether inside-out or outside-out) over cell-attached recording is that it provides the experimenter the ability to know precisely the transmembrane voltage, ion gradients and drug concentrations. While conditions on the pipette side of the membrane are usually held static, the conditions on the bath side of the pipette may be dynamically controlled as a way to study the regulation of P(open). As such, these patch-clamp configurations are ideal for the study of ligand-dependent channel activation. If ligands are applied slowly or have stationary effects once applied, data acquisition from these patch-clamp configurations can be treated the same as data acquisition from cell-attached patches.

However, the real strength of these recording modes comes from the ability to coordinate acquisition with solution changes made by an electronic valve controller or a piezo-electric switching device. This way, each patch can be used to establish both the control and the experimental response, as well as being used to follow non-stationary activity from the moment that a solution change is made.

To accomplish this, either the data acquisition should be controlled by the drug application system (DAS), or the drug application should be controlled from Clampex. In order to coordinate a single run of acquisition with the use of a DAS, a signal from the DAS can be sent either to the digitizer's Trigger In START input or to an acquisition channel. If a signal from the DAS is sent to the START input, then the START input must be appropriately configured in the Protocol Editor's Trigger tab. Once a start signal is received, acquisition continues until the Stop button is hit, a preset duration is achieved, or the disk space limitation is reached. The disadvantage to this approach is that no pretrigger data are acquired.

The alternative approach is to feed the signal from the DAS into one of the Analog IN channels configured as a trigger for variable-length event-driven acquisition. In this configuration, a pretrigger interval can be set. The system can then be armed by hitting the Record button. Data are not written to disk until a signal is received from the DAS. Acquisition can be terminated by the stop button, a preset duration, disk space limitation, or by the return of the DAS signal to a value below the trigger level.

A DAS may also be controlled in real time with either an Analog OUT or Digital OUT channel of the Digidata digitizer. In this case, the gap-free recording mode of Clampex may be used and configured to tag automatically the changes made to the output channel that provides the voltage-step commands to the DAS. Note that while a voltage-step command may be ideal to gate the function of a valve driver, it is not a suitable input to a piezo-electric solution switcher. Too abrupt a change in the command voltage to a piezo-electric device causes ringing in the motion, and may even damage the piezo-electric crystal stack. However, it is possible to control a piezo-electric solution switcher, such as the EXFO Burleigh LSS-3000, if the digitizer's voltage step command is conditioned by a simple RC circuit.

Gap-free recording containing periods of channel activity corresponding to agonist (or other drug) applications can be prepared for analysis by opening the file into a Clampex Analysis window and using cursors 1 and 2 to delineate segments of interest. File > Save As can be used to save each segment as an *.abf file for subsequent analysis in Clampfit.

If recordings were made in the variable-length event-driven mode, and the DAS step command was used as a trigger channel, when the saved data file is opened, it can be viewed as sweeps with View > Data Display > Sweeps. If the sweep-length exceeds the length of the longest triggered acquisition, then each triggered acquisition appears as a separate sweep. Data can then be sorted on the basis of sweeps and selectively saved.

Whole-Cell Recording of Responses to Drug Applications

The responses to drug applications in the recorded whole-cell mode normally take the form of macroscopic currents unsuitable for analysis as single-channel events. Rather, the responses are analyzed in terms of the peak amplitude and the kinetics of rise and fall. The Fixed-length events mode of event-triggered acquisition is ideal for this kind of data. The acquisition of the data as discreet sweeps provides the ability to detect, synchronize and average events. This mode is designed to capture any class of events that transpire within an anticipated (fixed) length of time. This mode creates a sequence of sweeps that can be averaged or analyzed for individual shape statistics in Clampfit.

It is worth noting the three acquisition characteristics that distinguish fixed-length acquisition from high-speed oscilloscope acquisition, although for post-acquisition analysis there is no difference:

- In Fixed-length events mode there is one sweep per event.

- Fixed-length events mode guarantees not to miss any events, whereas in High-speed oscilloscope mode multiple events can occur during a sweep and are missed as individual events.
- Most important, the trigger for each sweep in Fixed-length events mode can be uniformly coordinated with an external process such as a drug application system (DAS). For these kinds of data, it is probably most practical to coordinate the DAS and the acquisition with an external pulse generator. The pulse generator would send a signal in parallel to the DAS and the digitizer's Analog IN channel serving as the event trigger input. However, it is possible to use an Analog OUT signal from the digitizer for the same purpose. Using Clampex's Real Time Controls, if the Analog OUT signal is sent in parallel to the DAS and to the Analog IN channel serving as the event trigger, then data is written to disk only when a suitable Analog OUT signal is sent. The Analog OUT value can then be set back down and thus re-armed for another acquisition.

For example, with the data being read into Analog IN 0, and Analog OUT 0 used to control a valve driver, set the acquisition protocol for fixed-length event-driven acquisition with two Analog IN channels (0 and 1). Use a BNC "T" connector to send the Analog OUT 0 signal to both the valve controller and to Analog IN 1. In the Trigger tab of the Protocol Editor, select IN 1 as the trigger and a trigger threshold value below what is required for the valve controller. For example, if the valve controller needs a +5 V signal to activate the valve, set the trigger threshold for +3 V. Now the acquisition session can be armed by clicking the record button. When you want to activate the valve controller, you can type "5" into the text field for Analog OUT 0 and then trigger the valve when you press the <Enter> key. This should generate a response and a record of the relative timing of the valve control signal. To turn off the valve, type "0" into the text field and press <Enter>.

Whole-Cell Recording of Spontaneous Activity

While it is tempting to use Gap-free recording mode for events such as miniature synaptic currents, in fact, Fixed-length events mode of event-triggered acquisition is also ideal. Each event that passes a threshold triggers a sweep of uniform length, and while the sweeps are in a temporal sequence, they can also be overlapping in time. If multiple triggers are detected during the same interval, then the data are saved as multiple overlapping sweeps. This permits complete events to be saved independent of event frequency. Having the data in this format then permits quick analysis of shape statistics, such as amplitude and rise times, to be conducted in Clampfit, instead of having to first rerun Event Detection on the raw data in Clampfit to recapture the events for subsequent analysis.

Whole-Cell Recording of Evoked Activity

In most cases, fixed-length event-triggered acquisition is best for recording whole-cell evoked activity. It is possible to record multiple kinds of responses to stimuli in the event-triggered modes by setting up the trigger channel to trigger off an external device, such as a stimulus isolation unit that is used to shock a presynaptic nerve bundle. The pulse controlling the stimulator may also be read into Analog IN channel 0, while postsynaptic responses are acquired on Analog IN channel 1. If Analog IN channel 0 is selected as the trigger channel for fixed-length event-driven acquisition, data are acquired on channel 1 every time that the presynaptic nerve is stimulated. This way, sweeps of data on Analog IN channel 1 are recorded independent of any specific feature of the data, so that failures as well as evoked signals can be included in the analysis.

Alternatively, variable-length event-driven acquisition may be used if the data to be acquired is in the form of spike trains rather than single responses. However, in this case, the trigger should come from the data rather than the stimulus, so that complete runs of spikes are acquired.

Oocyte Recording

The use of Real Time Controls and gap-free recording is ideally suited for relatively low-frequency data acquisition, such as recording whole-cell responses from a *Xenopus* oocyte. If you recording the responses to bath-applied drugs, responses typically occur on the time scale of seconds, and so acquisition rates on the order of 10 Hz to 100 Hz are all that is needed. By double-clicking on the Scope window, you can call up the Window Properties dialog and select “Rolling display” to display the data as a chart record, suitable for acquisition on this time scale.

Bath application of drugs can be controlled in real time using the Digital OUT or Analog OUT signals. If an Analog OUT channel is used to control a valve driver, relatively accurate timing can be achieved by first typing the desired level in the Real Time Controls text field, and then waiting for a specific time mark (e.g. an elapsed time value) before hitting the <Enter> key. The Analog OUT only changes once the <Enter> key is struck. This way, a complete experiment containing multiple drug applications can be saved as a single file.

If you want to record a complete experiment in a single file but omit long times of inactivity such as when the drug is being washed out, Clampex can be toggled between the Record and View Only modes using the Pause-View toolbar button. Acquisition pauses, but the elapsed-time counter keeps counting.

Measuring Behavioral Data

Sometimes data need to be acquired during a behavioral experiment, where the animal may be unattended and inactive for long periods of time. An activity monitor or a photoelectric cell can be used on a trigger channel to control acquisition in the Variable-length event-driven mode, based on specific behavioral or environmental parameters.

6. Clampfit Features

This chapter is an introductory overview of Clampfit, outlining the main features and general organization of the program. Of special interest are the sections on event detection and single-channel analysis, introduced to Clampfit in version 9. Included in the single-channel analysis section are guides for Fetchan and pSTAT users upgrading to pCLAMP 10.

CLAMPFIT WINDOWS

Clampfit has the same standard Windows format as Clampex. Wherever possible, Clampfit uses the same commands and organizational principles as Clampex.

As in Clampex, all commands are included in the main menus, though many of these also have toolbuttons or can be accessed from right-click popup menus. Different menus are available according to the window type that has the focus (e.g. the Analysis window has no Format menu, while the Results window does), and the entries within menus of the same name similarly differ according to the selected window type.

Clampfit has seven window types:

- Analysis
- Data File Index
- Graph
- Lab Book
- Layout
- Results
- Statistics

These windows can be maximized, minimized, resized and tiled. Bring open but hidden windows into view by selecting them from the list at the bottom of the Window menu. Right-clicking in different regions in each window type brings up popup menus with options specific to the window, including in each case one that opens a Window Properties dialog. This allows users to configure the general appearance of the window, and these settings can then be saved as defaults (View > Window Defaults).

The basic features of each of the window types are set out in the following, but see also the online Help for greater detail.

Analysis Window

The Analysis window displays saved data files. There is no limit to the number of open Analysis windows that can be open at once. The data are displayed graphically, as traces, with subwindows for each signal within the file. A signal can be copied to a new signal within the same file, where it can be modified and then easily compared with the original (Edit > Create Duplicate Signal). Similarly, the results of a curve fit can be appended to a file as a separate signal (Edit > Create Fitted Curve Signal), as well as the idealized command waveform (Edit > Create Stimulus Waveform Signal).

Open files into Analysis windows from File > Open Data, or configure Clampex and Clampfit to automatically open newly recorded files in Clampfit (Configure > Automatic Analysis).   Configure the way files open into a new Analysis window or into an existent one, replacing the previous file, in File > Open Data Options.

Data can be displayed as Sweeps, or in Continuous or Concatenated modes, controlled from View > Data Display. In sweep mode you can choose to view any subset of the sweeps (View > Select Sweeps), and” toggle between viewing all the sweeps at once, a selected subset, or just one at a time (View > Toggle Sweep List). Step through sweeps one at a time with the “<” and “>” keys.

Many analyses can be performed on data in the Analysis window, including curve fitting, autocorrelation and cross-correlation, non-stationary fluctuation and V-M analysis. Data can be manipulated in a range of ways: subtracting a control file, averaging traces, adjusting the baseline, or averaging selected portions of the sweeps in a file to create an idealized sweep, to name a few.

Up to eighteen cursors are available to assist in making measurements and defining areas of the file for analysis. Cursors come in pairs, which you can add or remove from the View > Window Properties > General tab. Cursor pairs (cursors 1 and 2, 3 and 4, etc.) can be “locked” so they maintain position relative to each other as you move them about the file. Cursor text boxes display (optionally) time, amplitude and sample number, or, in the second of each cursor pair, delta values of these measurements relative to the first of the pair. Configure these and other cursor options by double-clicking on a cursor to open the Cursor Properties dialog box.

Since any number of Analysis windows can be open at time, fit data from multiple experiments can be directed to the same sheet in the Results window. In this way, fit parameters from control and treatment groups can be automatically tabulated and analyzed.

Selected parts of data files open in an Analysis window can be saved to new files, with the File > Save As > Options dialog. For example, you can save just the section of a file between cursors, or just the signals and parts of traces currently visible in the window.

Data File Index

The Data File Index (DFI) is a file management tool that allows you to construct an index of data files. Data files can be grouped together in separate DFI files, and within these sorted according to a wide range of parameters reported for each file. These options give you great flexibility in being able to rapidly find and organize files from particular types of experiments-especially valuable in managing large amounts of data from archival sources such as CDs or DVDs. Create a Data File Index from File > New Data File Index.

Graph Window

Graph windows display selected data in two-dimensional graphs. A range of graph types is available, such as scatter and line plots, and histograms. Any one graph can contain multiple plots, in which case a legend identifies these in the right-hand side of the window. Graphs can be named and the axes labeled in the View > Window Properties dialog, which contains a range of other configuration options as well. Axes for most graph types can be toggled between linear and logarithmic scaling with View > Axis Type.

A range of data manipulation options and analyses are available to apply to Graph window plots, including curve fitting, normalization and square root, amongst others. These are accessible from the Analyze menu when the Graph window is selected, and in the Graph window toolbar.

Graphs take their data from a Results window, and have no existence independent of their associated *.rlt (Results window) file. Graphs can thus be created from any Results window data, and when a graph is generated from some other source (e.g. a data file in an Analysis window with Analyze > Quick Graph > Trace vs. Trace), the data displayed in the graph is also written to an appropriate sheet in the Results window (for a Quick Graph, to the Quick Graph sheet). In consequence, to save a graph, you must save the Results window open at the time the graph is generated. If you manipulate data in the Graph window, for example, by normalizing it, the corresponding data in the Results window change to the new values resulting from the operation you have carried out in the Graph window.

Graphs are generated in Clampfit in a number of ways:

- From an Analysis window, the Analyze > Quick Graph command has options for generating I-V and Trace vs. Trace graphs. These graphs can be set up to update automatically for data files received from Clampex, in Configure > Automatic Analysis. They can also be dynamically linked to the file they are generated from so that, under cursor-dependent configurations of the graphs, moving the relevant cursors in the data file automatically alters the graph to reflect the new cursor positions.
- Histograms can be created from Analysis, Graph, and Results windows, with the Analyze > Histogram dialog. For data files in Analysis windows, these bin and count the amplitudes of the sample points in the file. For Graph and Results window data, you select the parameters for binning and counting.
- With a Results window selected, graphs can be created directly from a sheet in the window by selecting data in a column and using the Analyze > Create Graph command. The selected data are graphed against their row number on the X axis. Toolbuttons in the Results window allow you to select columns for both Y and X axes, and to add

further plots to the same graph. Alternatively, you are given many more configuration options if you use the Analyze > Assign Plots dialog.

- Several of the analyses in the Analyze menu (for the Results and Analysis windows) generate a graph as part of their output, e.g. the Kolmogorov-Smirnov Test, Autocorrelation and Nonstationary Fluctuation Analysis. In each case, the data in the graph is also written to a reserved Results sheet.
- With the Results window selected, graphs can be created with the Analyze > Event Analysis > Fast Graph dialog. This dialog is designed specifically for use with events data, which is usually found on the Events sheet. It can, however, be used to create graphs from any Results window sheet.
- The Event Detection > Define Graphs dialog allows you to create up to four graphs during an event detection session. The graphs are dynamically integrated with the other windows in the session, so that as new events are found they are plotted. You can select an event in all the windows linked within the session by clicking on the point corresponding to it in a Define Graphs scatter plot. For more information about this integration, see the *"Event Detection"* section later in this chapter.

Lab Book

Just as in Clampex, the Lab Book window is a text editor for logging events that occur while Clampfit is running. You can set how much you want written to the Lab Book with Configure > Lab Book Options, and add comments with Tools > Comment to Lab Book, or in a free-form text editor fashion directly in the window. The results of several Clampfit analyses, e.g. the Chi-square and Mann-Whitney tests, are recorded in the Lab Book.

There is always a Lab Book window open, called the System Lab Book. Copies of this can be saved to disk for editing or archiving elsewhere.

Layout Window

The Layout window is a page layout editor for formatting spreadsheets and graphs ready for presentation. Traces and curve fits can be copied into the Layout window from the Analysis window. Graphs and charts created in the Results workbook can also be pasted into the Layout window with additional text and draw objects added to create a high-quality figure. Files created within the Layout window are saved as *.alf files. Finished figures can be printed directly or copied and pasted as an enhanced metafile picture to other programs such as Corel Draw, Origin, or Microsoft Word or PowerPoint.

Results Window

The Clampfit Results window is similar to the Clampex Results window but with twenty tabbed data spreadsheets, in contrast to one in Clampex. Clampex's one cursor measurement sheet corresponds to the first "Cursors" sheet in Clampfit, and a saved Clampex results file can be opened directly into Clampfit onto this sheet (when the sheet is then labeled "Clampex").

Many analyses, including fitting, are available for data presented in the Results window, and all graphs have their data stored in Results window sheets. The results of many analyses are recorded in the Results window, often on sheets dedicated to results from that particular analysis. New results may either be appended to previous values or else, at the user's option, written over previous data. If data you are trying to append has different columns to that already present on a sheet, then it cannot be appended. In this case you are prompted to see if you want to replace the data currently on the sheet.

There are thirteen sheets that receive results from specific functions:

Table 6.1: Results sheets, and the functions from which they receive results.

Results Sheet	Function	For Window Type
Cursor	Tools > Cursors > Write Cursors & Append Cursors	Analysis
Events	Event Detection	Analysis
Bursts	Analyze > Event Analysis > Burst Analysis	Results
Statistics	Analyze > Statistics & Power Spectrum	Analysis
Basic Stats	Analyze > Basic Statistics	Results
Fit Params	Analyze > Fit	Analysis, Graph & Results
Correlation	Analyze > Autocorrelation & Cross-Correlation	Analysis, Graph & Results
Fluctuation	Analyze > Nonstationary Fluctuation	Analysis
Histogram	Analyze > Histogram	Analysis, Graph & Results
Power	Analyze > Power Spectrum	Analysis
Resistance	Analyze > Resistance	Analysis
V-M Analysis	Analyze > V-M Analysis	Analysis
Quick Graph	Analyze > Quick Graph	Analysis

In addition, there are seven numbered sheets (sheets 14–20) for general use. Data can be moved into the open sheets from the predefined sheets or from other programs using the Windows clipboard (i.e. by cutting or copying and then pasting in the new location). Data can also be copied into the open sheets from the Analysis window by using the Edit > Transfer Traces command. Statistics (*.sta) files opened in Clampfit transfer their data into a numbered Results window sheet as well as opening in a Statistics window.

As in Clampex, only one Results file can be open at any one time, though you can view this from multiple windows, for example, to compare two sheets, with the Windows >

New Window command. A new Results file is opened with File > New > Results. You can save a Results window as a *.rlt file. This saves any graphs associated with the window at the same time.

Statistics Window

Statistics files created in Clampex can be opened in Clampfit from the File > Open Other > Results & Statistics dialog. As well as opening into a Clampex-style Statistics window, the data are transferred to the first empty unreserved Results window sheet.

FILE IMPORT

The Windows environment in which Clampfit operates permits easy transfer of data and figures between Clampfit and other applications in the same environment. Simple cut and paste operations permit you to bring analytical results from other programs into the Results sheet. In fact, any text file can be opened into a selected Results sheet. Open the File > Open Other > Results dialog and select the text file you want to open. A dialog opens for you to select the column delimiter in the file and set other options, before the file is displayed. Old pSTAT events lists and histogram files can also be opened with the same dialog. These automatically open into the Events and Histogram sheets, respectively, without any configuration necessary.

Clampfit also imports numerical files for graphical display in the Analysis window. Any simple ASCII or binary output file can be opened into this window. When a non-Axon Instruments text or binary file is selected from File > Open Data, a conversions dialog opens. You need to complete this to convert the file into the scaled data that is displayed, specifying the number of signals present in the file and assigning signal names, units and scaling factors. A preview of the source file is provided, allowing you to specify a number of lines to be skipped so that header information is not misread as data. If the data file does not have a time column, you can specify the sampling rate, and Clampfit generates the time values when you import the data.

DATA CONDITIONING

Once raw data have been read into the Analysis window, Clampfit supports several kinds of data conditioning, including filtering and baseline correction. In addition to conventional high and low pass options, the filter functions (Analyze > Filter) include notch and electrical interference filters. See Chapter 8, *"Digital Filters"* for a detailed description of Clampfit filtering.

In addition to filtering, files can be baseline-corrected and spurious activity edited out. Analyze > Adjust > Baseline applies baseline adjustment as a fixed user-entered offset, or as an offset of the mean of the entire file, or of a particular region of the file. If there has been baseline drift in a sweeps file, for example, subtracting the mean of the first holding period can (depending on the file) bring each sweep to a common zero-value baseline suitable for further analysis.

Clampfit can also correct for linear drift in the baseline by applying a slope correction. Often the most useful form of baseline correction is manual correction. As shown in the figure below, the method for manual baseline correction permits the user to define the baseline in data which requires complex correction functions. When this option is selected, a line is drawn over the data. The user defines the appropriate correction by clicking to create inflection points, and dragging to where the slope of the baseline changes. This form of correction may be ideal for single channel data prior to export (see below).

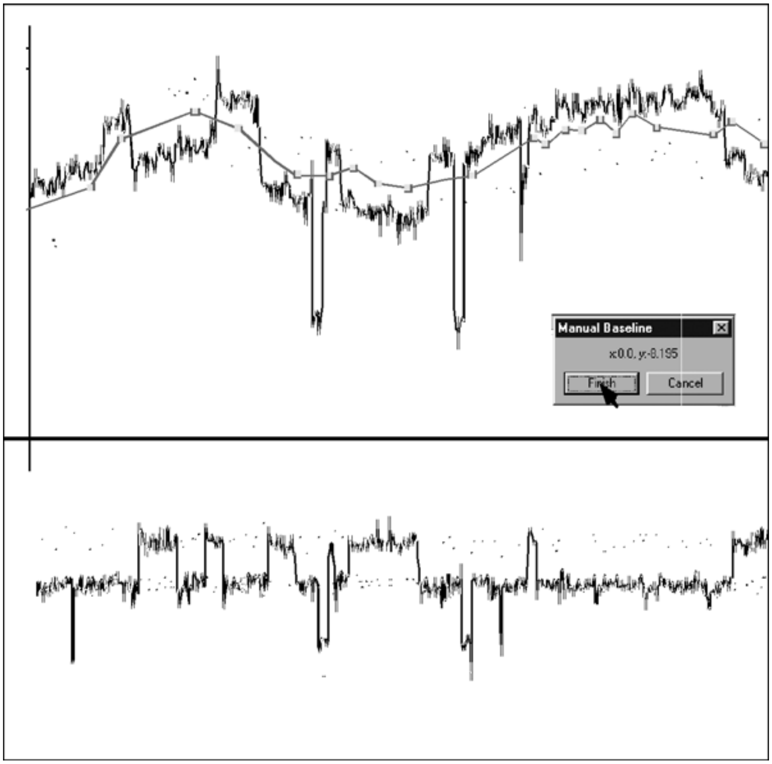


Figure 6.1: Manual baseline adjustment.

Another way of preparing a file for analysis is to remove the passive responses to the stimulation protocol. For this you can subtract a control file (Analyze > Subtract Control). If you run a stimulus protocol with lower voltages than those used in the actual test-in order to ensure you get a record of passive response without any cell activity-you can then multiply the resultant file by the appropriate factor before it is subtracted.

If you have no independently recorded control file suitable for subtraction, you can create a file of the passive responses from the test file. For a sweeps file where some sweeps contain only passive responses, you can identify these sweeps and use Analyze > Average Traces to average them, then save the resultant sweep. If there are no sweeps without any activity, you can select inactive sections from different sweeps and build up an average passive response for the entire sweep: use Analyze > Segmented Average.

You can also remove unwanted artifacts from files with Analyze > Force Values, or save specific sections of data files by using appropriate settings in File > Save As > Options.

Traces in a data file can be normalized prior to further analysis with Analyze > Normalize Traces. The dialog allows you to normalize the entire file or sweep, or to rescale the entire trace equally, but where only a selected duration maps to the zero-to-one range.

Any number of unmodified files, provided they are compatible (i.e. recorded under the same conditions, with the same acquisition mode, sample rate, number of signals etc.) can be combined into one file with Analyze > Concatenate Files. The files are ordered in their order of acquisition within the new file.

EVENT DETECTION

Event detection is a dynamically integrated mode of operation that binds the functionality of graphs, the Results window and other features specific to event detection, with the Analysis window where the file being searched is displayed. One data file at a time can be searched for particular trace features likely to mark a biological "event", such as a synaptic potential or a single ion channel opening. Three main types of event detection search can be carried out (accessed from the Event Detection menu).

- **Single-channel search**

  Single-channel records are translated into idealized "events", which are categorized as belonging to user-defined levels corresponding to different ion channel amplitudes in the patch. Clampfit's Event Detection > Single-Channel Search replaced "Events List" and "Latency" operation modes in Fetchan-pCLAMP's older single-channel analysis application.

- **Template searches**

  Templates are created by extracting and averaging segments of data that are manually identified as corresponding to an event. The template is then run through the data, identifying further events in the trace.

- **Threshold-based searches**

  Amplitude baseline and threshold levels are set, then the file searched for data that crosses the thresholds.

Event Detection Searches

Searches can be configured with a range of conditions to filter out false hits and to define events to ensure that event statistics-automatically taken for each event found-measure parameters in the desired way. Threshold and template searches can both have up to nine separate categories defined and concurrently searched for when a search is run.

All searches are able to accommodate high degrees of baseline drift provided this does not include abrupt shifts (in which case you are advised to first straighten the file with Analyze > Adjust > Baseline). As an additional aid to combat drifting baselines, levels can be manually changed while a search session is in progress. By dragging level markers in the Analysis window, events are immediately found in accord with the new levels, and statistics for these events recorded relative to the new levels. Levels are the only search parameters that can be changed on the fly. To change any other parameter in the search configuration you must stop the search, reconfigure the dialog and then press OK to restart the search.

When a set of event search parameters is confirmed (by pressing the OK button in the Search dialog) the event detection toolbar is inserted into the Analysis window containing the file being searched. Use this (or menu commands, or hot key keystrokes) to control the progress of the search. The first candidate event is highlighted in order that you can identify and assess it, with color-coded lines indicating its category (in a template or threshold search) or level (in a single-channel search). The program waits for you to accept the candidate event as a valid event, as a “suppressed” event (recognizably an event but with features such that you do not want to incorporate its statistics into the dataset), or to reject it entirely. Once you have concluded this, the next candidate event is automatically found and the process repeated. You are able to proceed in this manner for the entire length of the file (or for some defined portion of it) or you can choose to accept all automatically found events, and sit back and watch these accumulate. Once a search is finished you can configure a new search for the current file within the same event detection session, for example by repositioning cursors to search a different part of the file.

As the search proceeds you are able to view various types of events data, as they accumulate, in five other locations: the Event Monitor, Event Viewer, graphs, Results window, and Event Statistics dialog. Some of these are integrated so that selecting an event in one view brings it into view, highlighted, in other windows as well.

Event Monitor

Event Detection > Event Monitor is a small dialog that reports a few key statistics measurements for the current candidate event, providing a comparison with previously accepted events for that category (for peak-time events) or level (single-channel events). This is useful when assessing candidate events for inclusion into the data.

As each event is accepted, it is recorded, one line per event, in the Events sheet in the Results window. A range of measurements are always recorded, such as event start and end times, peak amplitude, rise and decay slopes, half-width, etc. for template and threshold events, and amplitudes and dwell times for single-channel events.

Graphs

Similarly integrated in the session, up to four graphs (Event Detection > Define Graphs) can be configured once search parameters have been defined and confirmed. Each graph can be configured as a conventional or logarithmic histogram or scatter plot, with all the measurements recorded in the Results window available for plotting. As with the Results window, these graphs dynamically update as new events are found.

Event Viewer

The Event Viewer (Event Detection > Event Viewer) contains a small Analysis window that accepts the trace segments identified as events in your search. These are overlaid one on another as sweeps within the window, and can be saved as an ABF or ATF file. Alone amongst the integrated event detection components, the Event Viewer can be kept open between event detection sessions and accumulate events from searches on different files.

Event Statistics

Lastly in the package of integrated data views, a summary of the statistics measurements recorded so far in any session can be displayed with Event Detection > Event Statistics. This opens the Event Statistics or Single-Channel Statistics dialog, depending on the search type, where means, standard deviations and data counts of all comparable events statistics taken in the session are reported.

For all these different "views" of events data, selecting an event in one view selects the same event in the other views. For example, clicking on a point in a scatter plot highlights the corresponding line in the Results window, highlights the event-marking lines on the trace in the Analysis window (and scrolls the window to bring it into view, if necessary), shows that event alone in the Event Viewer, and shows the values for the event in the Event Monitor.

Online Help has detailed discussion of all event detection functions, including an overview topic with links to all related Help topics. There is also a series of "How to" topics in the online Help for event detection. Additional information on single-channel event detection is also contained in the following section.

SINGLE-CHANNEL ANALYSIS IN CLAMPFIT

The following two subsections are written primarily as a guide to Fetchan and pSTAT users upgrading to the latest version of pCLAMP, but they should also benefit new users. They provide a general outline of where functionality from the old DOS single-channel analysis programs is found in Clampfit, as opposed to a precise mapping of command paths from one program to the other. It is assumed, for example, that the user is familiar with general navigation and data display commands within Clampfit, so these aspects of Fetchan and pSTAT are ignored.

Clampfit for Fetchan Users

File Subtraction

Use Analyze > Subtract Control to subtract a control file from a test file. The dialog allows you to scale the control file before it is subtracted. Note that subtraction can be set up to be applied automatically as files are imported into Clampfit, with Configure > Automatic Analysis > Subtract control file.

Any further baseline correction required must be carried out as a separate operation-use Analyze > Adjust > Baseline, where there are a number of options for this.

Clampfit has no direct equivalent to Fetchan's "Closest N Episodes" subtraction option. Subtracting an averaged trace and then applying suitable baseline adjustment can produce equivalent results. For example, subtracting the mean of a trace segment that has no activity in any of the sweeps (typically, the first holding will fit this criterion) brings a set of sweeps with rising or falling baselines to a common zero baseline.

Data Modification

The Parameters > General > Modify Data section of Fetchan has three settings.

- **Filter freq**

  Filtering is performed in Clampfit in Analyze > Filter, where you have a full range of filtering options. Unlike Fetchan, you must filter a file as a separate step prior to running event detection.

  Because filtering rounds out level transitions in the trace, it is taken into account in the calculation of event amplitudes and in the definition of "short" events used for automatic level updating. It is important that you ensure a record of all filtering that has been applied to the file is recorded in the data file header.

  Filtering in the amplifier at the time of acquisition, if telegraphed, as well as CyberAmp filtering and any post-acquisition filtering applied in Clampfit, are all automatically recorded in the file header. However, if you record without telegraphing the amplifier's filter frequency, you should set the Clampex Configure > Telegraphed Instrument setting to "Manual", and then enter the frequency information in the Lab Bench, which then writes it into the file header (see also *"Single-Channel Event Amplitudes"* on page 167 in Chapter 10, *"pCLAMP Analyses"*).

- **Derivative**

  A data trace can be converted to a plot of its differential values with the "diff( )" function in Analyze > Arithmetic.

- **Change polarity**

  Change signal polarity in Clampfit by changing the file scale factor. To do this, open Edit > Modify Signal Parameters and reverse the polarity of the scale factor. Note, however, that this can only be done with ABF files that have not already been modified in any way within Clampfit.

Analysis Mode

In Fetchan, different sorts of analysis are enabled by changing the operation mode in the Parameters > General > Analysis > Analysis mode field, which then shows the Analysis menu commands relevant to the selected analysis mode. Clampfit does not use this sort of organization, with all analyses being available once you have opened a file into an Analysis window. Clampfit functions replacing the Fetchan modes are listed below.

Episode Average

This Fetchan mode allows you to average all or selected sweeps from an episodic file. Use Analyze > Average Traces for this functionality in Clampfit.

To select specific traces to include in the average, move through the sweeps in the file using the ">" and "<" keystrokes, viewing each sweep highlighted against the remainder of the sweeps in the file. Press <Delete> for sweeps you do not want to include in the average (this hides the sweeps-they can be brought back into view with View > Select Sweeps). Once you have gone through the file and only have the sweeps you want to average in view, use Analyze > Average Traces, making sure that you have "All visible traces" for the trace selection, and "Display resultant trace only" checked.

You can save the resulting trace as an ABF file (File > Save As). In the Save As dialog, be sure to use the Data selection option, "Visible sweeps and signals" so that the newly saved file contains only the averaged sweep. This is because all the original sweeps are still in fact in the file, simply hidden from view. If a file containing these were used as a control file for subtraction, all the hidden sweeps as well as the averaged sweep would be used in the subtraction.

Segmented Average

The functionality in this operation mode can be found in the Analyze > Segmented Average dialog in Clampfit. The segmented average trace is built up in Clampfit in a similar way to Fetchan, with the user scrolling through the sweeps in the file in the upper window of the dialog, using cursors to define segments for addition to the accumulating average trace in the lower window.

In Clampfit, the entire sweep is always offered for creation of the segmented average, and the resulting trace saved from the dialog is for the entire sweep.

Pulse Average

Clampfit has no dedicated command for generating pulse averages, however this Fetchan function is easily duplicated. During a threshold event detection search, each event is copied to the Event Viewer, and left aligned to the start of the event. Once the file has been searched, press the Copy As button in the Viewer to save (and open) the assembled events as an ABF file. Then average the events with Analyze > Average Traces, creating the pulse average trace.

To recreate Fetchan pulse average behavior, set only one level in the threshold search, at the amplitude you would set the threshold if running pulse average in Fetchan.

Events List

This Fetchan operation mode identifies level transitions in gap-free and variable-length files. Event Detection > Single-Channel Search replaces it in Clampfit. Resulting data are written to the Results window Events sheet showing similar information to Fetchan EVL files.

In single-channel searches an idealized record is superimposed over the trace being searched, displaying the idealized events at each level. The duration of an idealized event is determined by the time the data crosses into and stays in the amplitude range of a defined level. The average amplitude of all the data points within that duration (less some points at the start and end of the event-see discussion of brief events below) is the amplitude of the idealized event. All the reported event statistics are taken from the idealized trace.

General configuration of single-channel searches-in Fetchan, set in Parameters > Special > Events List Analysis-is all carried out within the search configuration dialog in Clampfit. Set the baseline and channel open levels either by entering a numeric value in the dialog, or drag the level markers in the Analysis window. Note that you must set levels as close as possible to the amplitude reached when a channel (or multiple channels) is fully open, rather than at some threshold value part-way to this. The amplitude halfway between each level is in fact the threshold (50% crossing) for categorization of an event in either level. As in Fetchan, the baseline and/or other levels can be set to update automatically to follow an unsteady baseline, and you can also manually adjust levels during a search.

Filtering, which in Fetchan was performed at the same time as the creation of the idealized event trace, must be carried out independently in Clampfit, with Analyze > Filter.

For running the search itself, much Fetchan behavior is retained, although some of the commands (in the Event Detection menu, or use toolbuttons in the Event Detection toolbar, or keystrokes) have different names:

Fetchan	Clampfit
Include	Accept
eXclude	Suppress
iGnore	Reject
Undo	Undo
Nonstop	Nonstop

As each event is found in a search you can view key statistics for it, along with comparisons to previous events, in Event Detection > Event Monitor. If you are unhappy with the amplitude or duration automatically given to an event you can adjust this (as in Fetchan) by dragging the relevant idealized event-defining lines in the Analysis window.

Brief Events

Fetchan’s “short” events are called “brief” in Clampfit, and labelled “B” in the State column in the Events sheet. They can be optionally excluded from analyses you go on to perform on the events data.

Clampfit uses effectively the same algorithm for determining brief events as Fetchan. Since filtering rounds out level transitions in the trace, data points within two time constants of the start and end of an event are not used to calculate the amplitude (where the time constant is proportional to the amount of filtering applied). This means that if an event is shorter than four time constants (i.e. "brief") its amplitude cannot be calculated in the normal way. In these cases Clampfit reports the amplitude of the midpoint of the event as the amplitude of the entire event (see also *"Single-Channel Event Amplitudes"* on page 167 of Chapter 10, *"pCLAMP Analyses"*).

Latency

Fetchan's Latency mode is similar to the Events List mode, but applied to files generated under episodic stimulation where a stimulus was applied at the same time in each sweep. The time between the stimulus and the first response to this (i.e. the latency) is measured. In Clampfit this is handled in a normal Event Detection > Single-Channel Search, configured to ensure latencies are correctly measured:

1. Set cursors in the Analysis window about the sweep region to be searched, ensuring the first cursor is positioned at the time of the start of the stimulus. To do this you can reveal the stimulus waveform with Edit > Create Stimulus Waveform Signal and drag the cursor to the onset of the pulse, or use the Tools > Cursors > Cursor Properties dialog to position the cursor at the start of the epoch containing the stimulus pulse.
2. In the search configuration dialog, select the cursor pair you positioned above to define the search region.
3. Check the Latency Analysis checkbox in the search configuration dialog.

With the Latency Analysis checkbox checked, event start and end times reported to the Events sheet are measured from the start of the search region rather than from the start of the sweep. Cursor placement, as above, and selection of the cursors for the search region, ensure these values are measured from the time the stimulus was delivered. This allows the program to use the event start time for the first event in each sweep as the latency measurement. These values are used in the analysis of latencies (from the Events sheet) in Analyze > Event Analysis > Latency.

In addition to the steps outlined above, you can set an "Ignore initial duration" setting in the search configuration dialog. This is equivalent to Fetchan's Parameters > Special > First Latency Analysis "Ignore" setting, whereby level transitions for the stipulated period following the start of the search region are ignored. This is typically used to skip over capacitance transients that immediately follow the stimulus pulse.

All Points Histogram

To create a histogram for the amplitudes of all points in a data file, open the Analyze > Histogram dialog when the Analysis window containing the file is selected. All configuration of the histogram is done in this dialog. You set the number of bins here and can restrict the data to be included in it: by trace, by range within the sweep, and by data value.

Clampfit for pSTAT Users

pSTAT takes as its usual input the events list (EVL) files generated in Fetchan. Equivalent data in Clampfit are written to the Results window Events sheet, so this sheet should be selected in order to access analyses and graph options of the sort found in pSTAT.

Histograms

Much pSTAT functionality involves production and analysis of histograms, for example, of dwell times or amplitudes. In Clampfit, you can readily create simple histograms for any events measurements reported on the Events sheet with Analyze > Histogram:

1. Select the data column you want to display in a histogram.
2. Press the Histogram toolbutton in the Results window toolbar.
3. From the dialog choose the histogram type and bin width, and ensure that you have the correct Results window sheet and “All selected columns” in the Data section. You can restrict the data from the selected column so that you include only a specific row and/or data range.
4. Press OK to generate the histogram. The bin values are recorded on the Histogram sheet of the Results window.

By selecting more than one data column, the dialog allows you to include more than one parameter in a histogram graph (with their values combined or displayed separately, color-coded). You can also normalize the area under the histogram. The histogram function does not differentiate between data according to level or any other parameters, however. To do this, you need to use the Event Analysis Fast Graph (below).

Graphs

The Analyze > Event Analysis > Fast Graph dialog is a more powerful graph-generating dialog than Histogram, creating scatter plots as well as histograms. This highly configurable dialog is designed specifically for events data. It includes most of the options in the Histogram dialog, and also allows data selection by trace, search, and level, as well as by the value of a parameter different to the one being plotted.

Brief and/or suppressed events can be excluded from the plots. Where you have elected to plot events from more than one level, the plots are color-coded according to the level represented. Fast Graph is the general-purpose graph-generating dialog for all events data.

Fitting

Once a scatter plot or histogram is created, you have the full range of Clampfit’s fitting functionality in Analyze > Fit to fit a curve to this.

Besides displaying the fitted curve on the graph, fitting results are recorded on the Results window Fit Params sheet, and in the Analyze > Fitting Results dialog.

Frequency Analysis

Instantaneous frequency is automatically measured for all events as one of the basic measurements recorded on the Events sheet, as the events are found in an event detection search. You can use Fast Graph to display these data graphically if you wish.

pSTAT's Interval method frequency calculation is replaced by Analyze > Convert to Frequency:

1. Create a histogram of the "Event Start Time" values for the data, selecting a suitable bin width.
2. In the resulting histogram press the Convert to Frequency button to convert the count for each bin into a frequency, found by dividing the count by the bin width.

Burst Analysis

Burst Analysis in Clampfit is carried out from Analyze > Event Analysis > Burst Analysis. The burst analysis provided in Clampfit differs in a number of respects from that in pSTAT. There are two basic means of finding bursts in Clampfit:

- Set an inter-event interval. Events found to be closer than or equal to this interval are classified as belonging to the same burst. The numerical value of Clampfit's inter-event interval is the same as the inter-burst interval that would be used in pSTAT, the difference being that Clampfit looks for closed-channel times less than the interval in order to cluster events together as belonging to a single burst, while pSTAT looked for closed-channel times greater than the interval, in order to separate bursts.
- Use "Poisson surprise", which measures the degree to which one is "surprised" that a frequency of occurrence deviates from a Poisson distribution.

You can elect to search for bursts of events belonging to a particular level, or for bursts of all levels. These all-level searches can be "merged", so that a sequence of nonzero-level events is treated as one event.

A range of data about the bursts found in an analysis are reported to the Bursts sheet, including the number of events in each burst, the burst duration, the mean intra-burst interval, the event frequency and the mean open time of the events. Histograms of these properties can be generated in the usual way.

See also *"Burst Analysis"* on page 174 in Chapter 10, *"pCLAMP Analyses"*.

P(open) Analysis

Analysis of the probability that a channel is open is performed with Analyze > Event Analysis > P(open). This analysis reports the number of events included in the calculation, the total time range, and, on the basis of the way in which you configure the dialog, the probability of a single channel being open and the probability of any channel being open.

The dialog also gives the option of generating a P(open) histogram similar to pSTAT's. As in pSTAT, you can manually enter an interval setting for this, or have it done automatically by Clampfit.

See also *"$P_{(open)}$"* on page 176 in Chapter 10, *"pCLAMP Analyses"*.

Latency Analysis

Latency analysis (Analyze > Event Analysis > Latency) is the measurement of the times from the application of a regular stimulus in each sweep until the start of the first event in the sweep. Optionally, you can create a histogram of the latencies from this dialog.

Events data need to have been collected under the right conditions in order that this analysis can be carried out. The latency values reported are simply the event start time (from the Events sheet) for the first nonzero-level event in each sweep. To ensure that this value is a measure of the time from the onset of the stimulus, you need to have configured the single-channel search (Event Detection > Single-Channel Search) appropriately. This involves checking Latency Analysis in that dialog, and setting the event detection search region to start at exactly the time of the stimulus.

See brief instructions above (*"Clampfit for Fetchan Users"* on page 94) or more detailed instructions in the Online Help.

Peri-event Analysis

This analysis is new in Clampfit, with no equivalent in pSTAT. It measures the intervals between selected events and events that occurred within a stipulated time range of these. Events are selected as the central events for this analysis by tagging them. This can be done within a search session with Event Detection > Accept and Tag or Event Detection > Process Selection, or by typing "T" into the State column in the lines of the selected events on the Events sheet.

See also *"Peri-event Analysis"* on page 175 in Chapter 10, *"pCLAMP Analyses"*.

Idealized Trace

Clampfit displays an idealized trace showing dwell times and average amplitudes for each event, color-coded for the event level, when events are being found in a single-channel search. Once the event detection session is closed, the idealized trace cannot be re-created.

Curve Fitting for Single-Channel Results

Clampfit provides the Gaussian function for fitting to amplitude data and exponential probability functions for fitting to either conventionally binned dwell-time data or dwell-time data that have been transformed by log interval binning.

Fitting to dwell-time data supports either least-squares minimization or log likelihood maximization. However, it is not recommended that conventionally binned data be fitted using maximum likelihood estimation, which is primarily recommended for fitting to log-binned data. Clampfit provides an efficient and accurate algorithm for maximum likelihood fitting to log-binned dwell-time distributions, using the EM algorithm for initial parameter estimates and the variable metric fitting method to fine-tune the fit.

For example, Figure 6.2 (below) shows variable metric fits of 2, 3 or 4 exponential terms to log-transformed open dwell-time. The fit improves with the higher order functions.

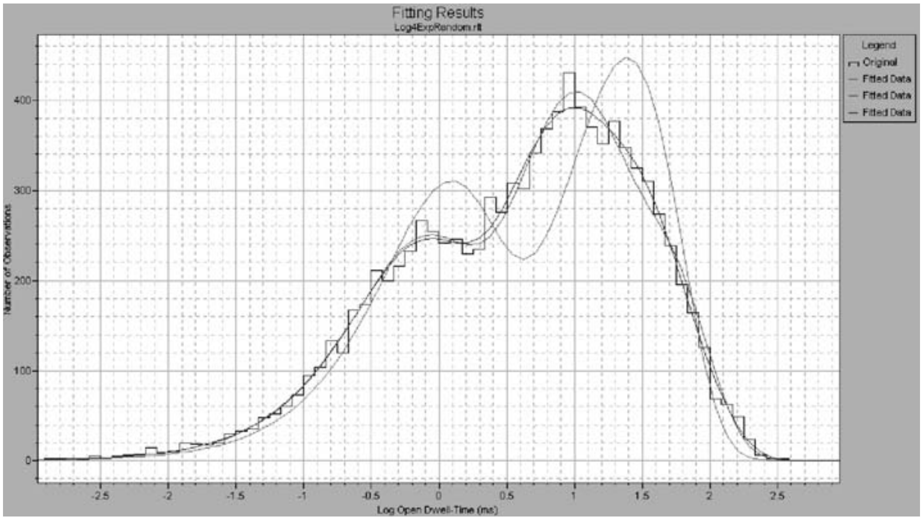
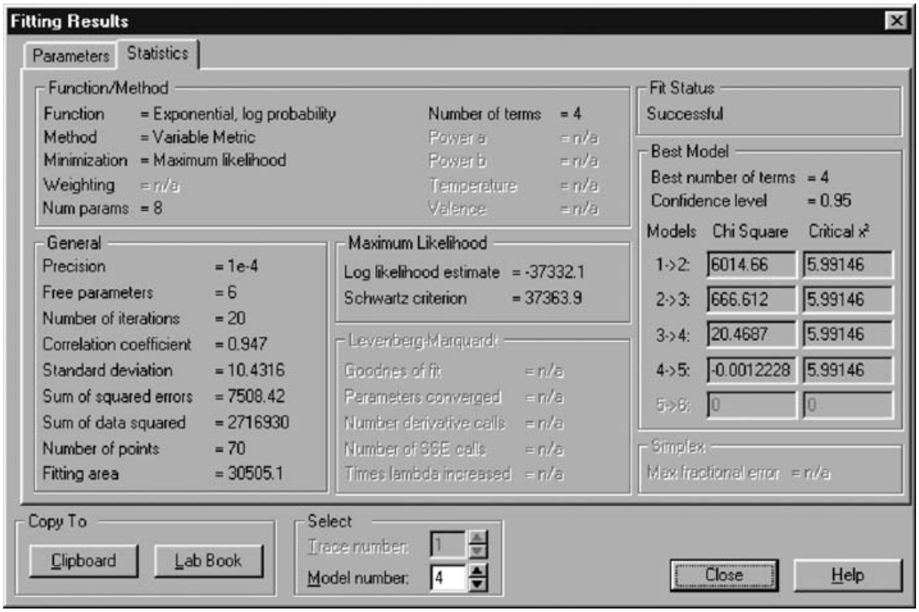


Figure 6.2: Variable metric fits with 2, 3 and 4 exponential terms.

Square root transforms of the binned data can be calculated using Analyze > Square Root column arithmetic in the results sheet. However, Clampfit does not provide options for excluding zero-count histogram bins from the fit nor does it correct for binning and sampling promotion errors of conventionally-binned data, as does pSTAT. Even so, you might find that Clampfit provides a more flexible environment for fitting, display and presentation of dwell-time and amplitude histograms, fitting residuals and component curves.

Clampfit also provides an option for statistically determining the best fitting model (number of terms) for dwell-time or amplitude distributions; check Compare Models in the Analyze > Fit > Data/Options tab. Fitting models are iterated automatically. Moreover, you can specify confidence intervals for standard and maximum likelihood comparison statistics. The respective F-statistic or Chi-square table values are computed automatically and displayed along with the test statistics so you can evaluate the results without referring to statistical tables. This information is provided following the fit in the Analyze > Fitting Results > Statistics tab. For example, Figure 6.3 shows the results of an automatic model comparison for the log-binned data shown above:



Function/Method

Function = Exponential, log probability
Method = Variable Metric
Minimization = Maximum likelihood
Weighting = n/a
Num params = 8

Number of terms = 4
Power a = n/a
Power b = n/a
Temperature = n/a
Valence = n/a

Fit Status

Successful

Best Model

Best number of terms = 4
Confidence level = 0.95

Models	Chi Square	Critical x²
1->2:	6014.66	5.99146
2->3:	666.612	5.99146
3->4:	20.4687	5.99146
4->5:	-0.0012228	5.99146
5->6:	0	0

General

Precision = 1e-4
Free parameters = 6
Number of iterations = 20
Correlation coefficient = 0.947
Standard deviation = 10.4316
Sum of squared errors = 7508.42
Sum of data squared = 2716930
Number of points = 70
Fitting area = 30505.1

Maximum Likelihood

Log likelihood estimate = -37332.1
Schwartz criterion = 37363.9

Levenberg-Marquardt

Goodness of fit = n/a
Parameters converged = n/a
Number derivative calls = n/a
Number of SSE calls = n/a
Times lambda increased = n/a

Simplex

Max fractional error = n/a

Select

Trace number: 1
Model number: 4

Figure 6.3: Fitting Results dialog Statistics tab.

The model comparison statistics are shown to the right in this information dialog. Note that since the difference in the Chi-square value for models 4 and 5 (that is, 4 and 5 terms) is not significant, the 4-term model is accepted as the best fit.

QuB

Clampfit provides an option to export files in *.ldt format. Single-channel data exported in this format can be analyzed with QuB, a specialized suite of applications available from the University of Buffalo. These programs provide an alternative to Clampfit for creating idealized records by applying maximum likelihood interval analysis to find the most likely rate constants for specific models. Information about these programs and downloads can be found on the QuB web site (http://barrel.med.buffalo.edu/). Note that these programs represent a highly specialized approach to single channel data, referred to as hidden-Markov modeling, and are neither directly supported nor endorsed by Axon Instruments. Therefore, Clampfit users are advised to read the information and the references provided on the QuB web site before using the QuB programs to analyze their data. It should also be noted that the QuB methods are best suited for data that have not been heavily filtered.

Unlike the idealization of single channel records performed by Fetchan, the idealization procedure in the QuB suite does not permit the identification and editing of individual events. Therefore, the data must be carefully preprocessed in order to assure that they are suitable for idealization. This can be done either in Clampfit or in the preprocessing module of QuB.

Preprocessing in Clampfit is easy. First, data to be analyzed by QuB should be baseline-adjusted before export, best accomplished using the manual adjustment method described above. If a record has segments of noise from the breakdown of patch resistance or other spurious signals, these portions of the record can be removed before export by using the "blank" command. The recording is then exported as multiple segments, and the QuB programs can be applied to fit a model that would describe the activity within segments (information about time between segments will be lost in the QuB analysis). If a recording clearly contains multiple kinds of channel activity (distinguishable for example, by amplitude), the "blank" command can also be used to create a sequence of homogeneous segments, suitable for fitting to a single model.

Note that, along with the hidden Markov model analysis, the SKM and MIL modules of the QuB suite do generate amplitude and event duration histogram files, respectively, which can be opened directly as Results sheets in Clampfit. Once the X-axis data in the QuB duration histograms is converted to Log values (using Analyze > Column Arithmetic in Clampfit), the SKM-idealized data can be fit with conventional methods, as described above. This approach permits the direct comparison of QuB and Clampfit idealizations.

FITTING AND STATISTICAL ANALYSIS

Clampfit boasts a wide range of predefined fitting functions for data displayed in Analysis, Graph and Results windows, or you can define customized functions. The Analyze > Fit dialog allows you to select the fitting method and configure a range of parameters relevant to the fitting function and method you select. While fitting in the Analysis and Graph windows allows the fit to be displayed along with the data, fitting in these windows is only suitable for simple X-Y pairs of data. Data in a Results window, on the other hand, can be fit with complex functions such as the Goldman-Hodgkin-Katz equation by assigning multiple variables to specific columns.

Fitting in Clampfit is dealt with thoroughly in Chapter 11, *"Curve Fitting"* and Chapter 12, *"Fitting Functions"*.

A powerful range of parametric and non-parametric statistical analyses are available to apply to data in the Analysis and Results windows. For analyses carried out on data files in the Analysis window you can identify specific epoch regions to which to apply the analysis from within the analysis configuration dialog. Alternatively, if you don't want to analyze the whole file, set a cursor pair about the region you want to analyze before you open the configuration dialog, then select this region after you have opened the dialog. To select Results window data for inclusion in an analysis use the Select Columns dialog opened from the analysis dialog, or the Analyze > Extract Data Subset dialog for datasets based on conditional settings.

Other analyses (all located in the Analyze menu) are:

- **Burst Analysis**: finds closely packed groups of events in single-channel and peak-time events files and reports a range of statistics for these.
- **Latency:** measures the time between the onset of a stimulus and the response to this, in episodic files (for single-channel and peak-time events).
- **Peri-event Analysis:** measures the intervals between selected events and events that occurred within a stipulated time range of these (single-channel and peak-time).
- **P(open):** analyzes single-channel data to calculate the probability that an ion channel is open.
- **Kolmogorov-Smirnov Test:** a non-parametric method used to assess the probability that data from two distributions belong to the same population.
- **Nonstationary Fluctuation:** computes the mean and variance for all the sweeps at each sample point within a selected area of trace activity and subtracts mean and variance values derived from a baseline section of the trace.
- **V-M Analysis:** generates a plot of variance against mean for data obtained under different experimental conditions to evaluate transmitter release.
- **Autocorrelation and Cross-correlation:** available for Results, Analysis and Graph windows data, these are time series analyses that look for cycles within the selected dataset, either by comparing time-shifted data with itself, or to data from a different file.

CREATING FIGURES IN THE LAYOUT WINDOW

Data reduction and presentation often involve many steps, from simple fits of the raw data to extrapolation of complex functions based on those fits. Clampfit is configured with a wide range of predefined fit functions to get both the raw data and final analyses into the same figure. For example, concentration-response data can be fit to the Hill equation, or reversal potential data from ion substitution experiments can be fit to the extended Goldman-Hodgkin-Katz equation. The Layout window can then be used to format a presentation of both raw data and plots of the final analysis.

The Layout window provides a place where the many elements of Clampfit come together for final preparation and presentations. Graphs, raw data traces, and even tabulated data can all be copied directly into this window. To enter statistical results from the Lab book, first copy and paste them into the Results window, and from there, copy them into the Layout window.

While Layout window files can be saved, they can only be reopened within Clampfit. Therefore, to transfer figures to another program, you need to use the standard Windows Copy and Paste functions. Using the option of Paste Special - Enhanced Metafile provides the best results. Otherwise, first copy and paste the figures into a Microsoft application such as Word, and from there, transfer them to the target application.

7. Clampfit Tutorials

The scenarios presented in this chapter introduce Clampfit functionality by implementing procedures that are frequently used in the evaluation of physiological data. Occasionally the tutorials do not take a straightforward approach, in order to introduce you to a wider variety of features than would otherwise be the case. Once you master these features you may wish to perform the analyses in a more direct manner.

The scenarios highlight some of the features discussed in the last chapter. We will give menu commands for each step in the analysis, but you should also become familiar with toolbuttons and hotkeys. Hold the cursor over a toolbutton to read its tooltip, and hotkey combinations are reported beside menu commands. Also try the context-sensitive commands in the popup menu that a right mouse click opens. Context-sensitive online help is available everywhere in the program by hitting the <F1> key or through the Help buttons in the dialogs.

The sample files are found in the ...\Program Files\Molecular Devices\pCLAMP10.0\Sample Data folder. When following the tutorials, you might want to save the changes from time to time. This is not explicitly mentioned, and is not really necessary, since you will find that most of the steps are reproduced quickly. If you want to save the analyzed files nevertheless, we recommend not replacing the original sample files. pCLAMP handles two different types of binary files. Integer ABF files are assumed to contain original data and cannot be overwritten from within the programs. Analyzed files are by default saved in floating point ABF format, and can be overwritten. Overwrite only ABF floating point files.

Clampfit can be used to manipulate data in illegitimate as well as legitimate ways. Remember that it is your responsibility as a scientist to consider carefully the changes you introduce into the data and to maintain the integrity of the results.

CREATING QUICK GRAPHS

Clampfit can generate current-voltage plots with just a few mouse clicks. Furthermore, in cooperation with Clampex, you can create I-V plots automatically, in real time, immediately after the application of a step protocol. This is explained in *"Real-Time I-V Plotting in Clampfit"* on page 76).

A Quick I-V from Whole-Cell Currents

Use File > Open Data to open the sample file cav1.dat. It is a whole-cell patch clamp recording showing outward currents in response to a voltage step protocol. To see the protocol that was applied, select View > Stimulus Waveform Display, or press its toolbutton in the main toolbar. From a holding potential of -50 mV, depolarizing voltage steps were applied using the Analog Output Channel AO #0. We will plot the peaks of the elicited outward currents versus the step voltage:

1 Set cursors 1 and 2 to 32 ms and 250 ms, respectively. You can do this by dragging the cursors to their new positions, or by double-clicking the cursor's text box and entering the X axis value in the Time field.

2 Select Analyze > Quick Graph > I-V, or click the toolbutton in the Analysis window toolbar.

3 There are two ways to assign the step voltage to the X axis: specify it by the positions of cursors 1 or 3, or define it using epochs. In the example all four cursors are placed in Epoch B, so each of the three Waveform options in the X Axis group leads to the same result.

4 On the Y axis, we are going to plot the Peak outward currents elicited during the voltage step. There is only one Signal (AI #0), which we can specify. To avoid the capacitive transient at the onset of the voltage step, we restrict the search for a Peak of Positive Polarity to the Region between Cursors 1..2.

5 Clampfit offers the opportunity to smooth the signal while Peak is selected, minimizing the influence of high-frequency noise, or individual channel openings in relatively small cells. Up to 21 smoothing points are allowed, but you will have to determine the appropriate choice depending on the signal and sampling interval. To do so, either compare I-V curves with and without smoothing enabled, or smooth the signal using Analyze > Filter. In the third tutorial we will learn how to compare a filtered signal to the original.

6 For our comparatively slow currents, 11 Smoothing points provide a sufficiently smooth signal with little danger of over filtering.

7 Click OK.

Clampfit creates a new Graph Window with the I-V plot, which can be customized in many respects after double-clicking to obtain the Properties dialog, or by using the items in the View menu.

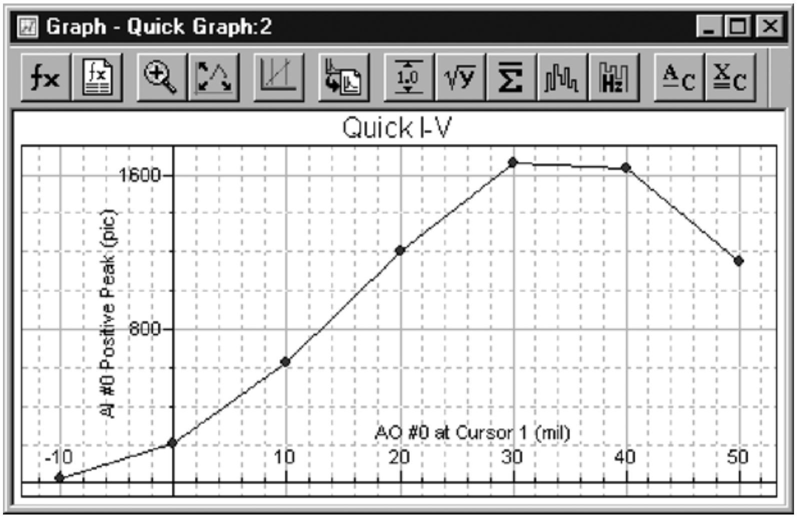


Figure 7.1: Quick I-V graph for cav.dat.

A Quick I-V from a Cell-Attached Recording

In cell-attached and inside-out patch clamp recordings, the recorded current must be inverted (see Penner 1995 for a discussion on polarity). The membrane potential can be calculated by simply inverting the command potential in the inside-out configuration, while in the cell-attached mode the resting potential of the cell has to be added. Many experimenters invert the current and the command potential using a negative scaling factor during the acquisition. Since in cell-attached recordings the computation of the membrane potential requires knowledge about the resting potential, other experimenters invert the current only, while a third group acquires both in the original polarity. Clampfit deals with the different strategies by giving you the opportunity to invert either of the signals.

Use File > Open Data to open the sample file oncelliv.abf. It is the current response to a voltage step-protocol applied to a cell-attached patch with many copies of a nonselective cation channel almost steadily open. The current was inverted using a negative scaling factor, which you can review by double-clicking on the Y axis header. On the Output tab you see that the command potential stored in the file was not inverted, so we have to do that now.

We are going to plot the mean current during the voltage step versus the step voltage

1 Double-click on cursor 1 and enter 12.5 into the Move To > Time field.

2 Click Next to move to the Properties of the next cursor.

3 Select the Epoch radio button, and choose Epoch C Start from the Epoch list box.

4 Click OK.

5 Open the Analyze > Quick Graph > I-V dialog.

6 In the X axis section select the Epoch radio button and choose Epoch B level from the Epoch list.

7 Enable the Invert checkbox in the X axis section.

8 In the Y axis section select Cursors 1..2 from the Region list and select Mean.

9 Ensure Dynamically link Quick Graphs to Analysis Window is checked at the top of the dialog

10 Click OK.

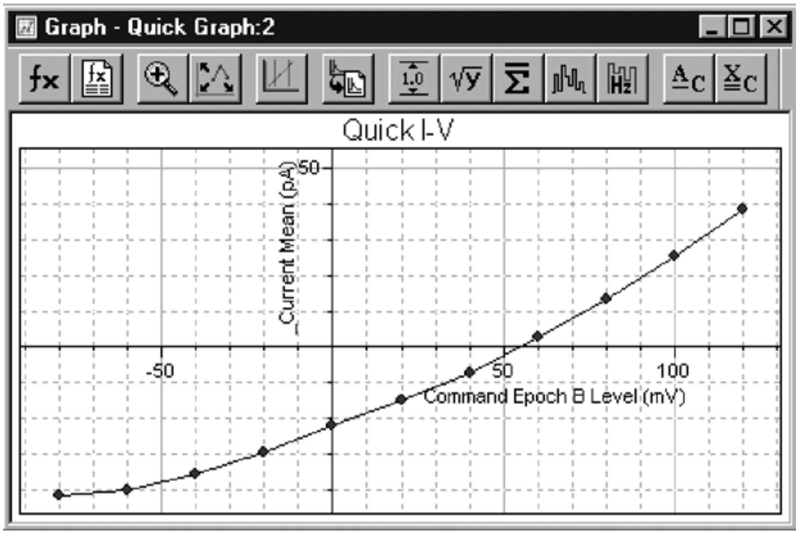


Figure 7.2: Quick I-V graph for oncelliv.abf

Tile the Graph and the Analysis windows vertically using Window > Tile Vertically and move cursor 1 a little using the mouse or the keyboard arrow keys. The I-V plot is automatically updated to the mean between the new cursor positions. From other experiments it is known that this channel type reverses around 0 mV. Since the current in our recording reverses at 55 mV, we can assume a resting potential of the cell of -55 mV. Select Analyze > Adjust > Stimulus Waveform and enter 55 into the Analog OUT #0 field.

Since it is relatively unusual that the resting potential can be precisely determined and corrected for, Analyze > Adjust > Stimulus Waveform only applies to the currently analyzed file and cannot be saved. Should you want to save the waveform adjustment to correct for an offset in the command potential during the recording, go to Edit > Create Stimulus Waveform Signal and use this signal for the evaluation.

Quick Trace vs. Trace Plots from Whole-Cell Currents

The application of voltage ramps frequently contributes to the characterization of ligand-gated currents (Okada *et al.* 1994, Tang & Papazian 1997). The sample file inrctc02.dat is an example for an inwardly rectifying current in response to a voltage ramp.

Select Analyze > Quick Graph > Trace vs. Trace. This function allows plotting different elements of a data file versus each other sample by sample. The dialog provides two identical sections for the definition of the X and Y values, respectively. The available Traces, Waveforms, and Signals are selectable from drop-down lists. Additionally, as in Quick I-Vs, the values can be inverted to account for data from cell-attached or inside-out recordings.

We want to plot the only available sweep of the input signal AI #15 versus the output signal AO #0, so the default settings apply-Sweep #1 of waveform AO #0 (mV) for the X axis, against Sweep #1 for the signal AI #15 for the Y axis. While in cav1.dat we had to avoid the capacitive transients at the beginning of the voltage steps, we now can perform the analysis on the entire ramp epoch. Therefore, specify “Epoch C - Waveform 0” in the Region drop-down list.

Clampfit creates a plot that, in fact, looks similar to the original data file, but is calibrated in mV and pA, respectively, and does not include the holding sections and artifacts prior to or following the actual ramp waveform. To compare with the current elicited under control conditions, open the file inrctc01.dat and repeat the previous steps, making sure that Append is selected in the Destination Option group.

Preparing Data for Presentation

We are next going to create a Layout window containing the original data files, the voltage step protocol, and the I-V curves:

1. Select the Analysis window that contains inrctc02.abf.
2. Edit > Copy to Layout Options allows you to choose between copying analyzed data files either in the fashion they are displayed in the Analysis window, or using the Page Setup settings. In addition, you can specify three parameters to be automatically included in the comment line: the file name, the acquisition date and time and the comment.
3. Select Page Setup settings and uncheck the three parameters for the comment.
4. Select Edit > Copy to Layout Window. Clampfit opens a Layout Window and prompts where to place the graphic within a virtual grid on the canvas, and which comment to add.
5. To illustrate the voltage ramp that was applied, activate one of the Analysis windows, select View > Stimulus Waveform Display, scale and then click the “==> Clipboard” button. Select View > Layout Window and paste the waveform.
6. Return to the Graph Window we created before. The Trace vs. Trace Graph can be customized in the Properties dialog that is brought up by double-clicking, or through

the right-click menu or if you select View > Window Properties. For the sample Layout above, starting from the factory default window properties, all fonts and the axis titles were altered, and the grid lines, the frame box and the legend title were removed. When the graph suits your needs, copy it to the Layout Window. You now have a figure illustrating the activation of an inwardly rectifying current by drug application.

PRECONDITIONING NOISY SINGLE-CHANNEL RECORDINGS

In electrophysiology, the best strategy to deal with noise is to avoid recording it. However, you will often have important data that cannot be analyzed without preconditioning. To learn more about Clampfit's functionality during this tutorial, we will introduce artificial noise into a data file. Then we will remove the noise, while maintaining as much of the information in the file as possible.

The sample file kchann.dat is a patch clamp recording of a single active channel. It is the kind of recording often presented as "typical data" in publications. That means, hardly ever any recording is as clean as this one, so we are going to make it a little more realistic. We will work with a section of the file only:

1. Select View > Data Display > Sweeps and enter "20000" in the Sweep length field of the dialog box.
2. Now select the first sweep using the View > Select Sweeps command, or right-click in the data area for a popup menu option.
3. We use Edit > Transfer Traces to convert the binary data file into ASCII data in the Results Window. Edit > Transfer Traces gives you the option to transfer the data from the Analysis window to the Results window or to a Graph window. Make sure Results window is selected.
4. The maximum number of samples Clampfit can transfer is 1,000,000. The sweeps are 20,000 samples long, so we can select Full trace in the Region to transfer list.
5. We want to transfer the first sweep only, so the Trace Selection group should report All visible traces.
6. Press OK and look at the Results window to find the first portion of the data on Sheet 14.

Now we can start creating the artificial noise. We first generate line frequency hum:

1. Select column C by clicking on the column header and then go to Analyze > Create Data, or alternatively press the toolbutton in the Results window toolbar.
2. The Create Data dialog allows you to define the Fill Range in the From and To fields for columns and rows. The default setting is the current selection on the Results sheet, which is rows 1 to 20,000 of column C in our example. We can keep this default here.
3. The Series Option group offers a set of predefined functions plus the Expression field for custom formulas.

4 In the Data Limits fields the values of variables used to generate the various predefined series can be specified. Click on different functions in the Series Option list to see which variables can be defined.

5 If the Fill Range comprises more than one column and row, Fill Direction determines the way in which Clampfit fills the cells. Your selection is represented graphically in the dialog.

6 Since we are going to create a sine wave, we use a custom expression function, which must be specified in terms of a function with the independent variable *x*. Let's use a line frequency of 50 Hz, corresponding to a 20 ms wavelength. Therefore, we will use a variation of the function:

$$y = sin(2\pi \times time / wavelength) =$$

$$sin(2\pi \times time / 20\ ms) =$$

$$sin(2\pi \times time / 10\ ms) =$$

The constant π is accessed by typing in "pi". The amplitude of the sine wave will be about 1/4 the amplitude of the signals, which is almost 16 pA. So the expression we enter is:

*4 * sin(pi * x/10)*

7 On the Results sheet, we find the time, in sampling interval steps, in the first column. Since this is the independent variable, we enter 0.1 into the X Increment field. The X Start Value can be any number in this example, since it only introduces a phase shift in a periodic function like a sine.

8 Click OK.

9 We additionally create harmonics in columns D-F, reducing the wavelength (10) in the term by integer division and the amplitude (4) in integer steps. To keep track which signal you created in which column, rename them by double-clicking on the header of column C, always pressing Next after you have typed in a convenient name.

10 Now that we have the original signal in column B, and line frequency hum in columns C–F, we are going to add the components together. Select Analyze > Column Arithmetic.

11 In the Column Arithmetic dialog there are several controls that assist in entering an expression. The Columns button allows you to specify a list of columns using the "cA" notation explained in the dialog, or to Select from the list. The Operators button opens a drop-down list of basic mathematical operators, while the Function button lists the functions supported by Clampfit. See the online help for a more detailed description. The items in the Special drop-down list yield the row number, the maximum number of rows in the columns the expression includes, and the constant π. Undo reverts the most recent change you did to the expression. The Specify Region group allows restricting the row range the expression is applied to. When only a subset of the rows is to be analyzed, the Excluded region is forced to zero if this option is checked, else it remains unchanged.

12 We want to add all but the Time column and put the results out in an empty column. So the expression you should apply is:

$$cG = cB + cC + cD + cE + cF$$

13 Both OK and Apply execute the expression, but after Apply the dialog remains open. Now the data in column G should be a patch clamp recording with a lot of line frequency hum. We are going to save this composite signal, open it as data and remove the noise.

14 Select File > Save As, set the file type to Axon Text Files (*.atf) and save with a name like kc_noise. Then select File > New > Results to close the ATF file in the Results window.

15 Open the Results we just saved into an Analysis window with File > Open Data (again set the file type to *.atf).

16 Clampfit by default assumes the multi-column ATF file to be a sweep type file and displays each column as one sweep. Select sweeps 1 and 6 only using View > Select Sweeps and go to View > Distribute Traces.

17 Distribute Traces allows you to offset the display of overlaid traces for a user-defined distance. Note that the offset is only added to the display, not to the data. Enter 30 in the Distance field.

18 Select Autoscale All Y Axes from the right mouse popup menu.

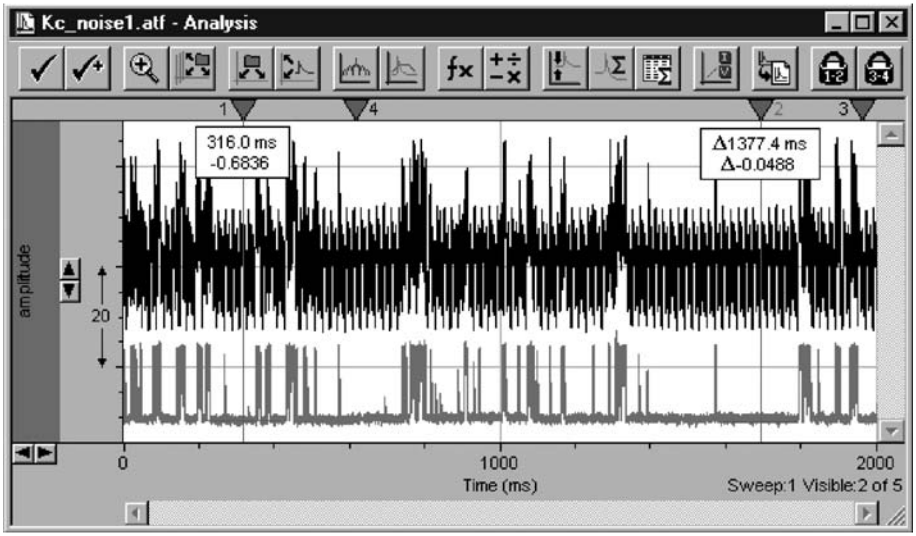


Figure 7.3: Analysis window after Distribute Traces.

19 The original recording, displayed in the lower trace, is now heavily contaminated with noise, as can be seen in the upper trace. Since in real life you would not have created the noise yourself, it is useful to perform a power spectrum before filtering:

a Select sweep #6 only, autoscale the Y axis once more and go to Analyze > Power Spectrum.

b A number of settings can be adjusted in the Power Spectrum dialog, which are explained in detail in Chapter 9, *"Clampfit Analysis"*. For our purpose a Rectangular Window together with the maximum Transform Length will do. There is no need to Exclude spectral bins, and the analysis will apply to the full length of the Active trace. Clampfit creates a Graph Window with the power spectrum in a Log-Log plot.

c Right-click on one of the axes or go to View > Axis Type and select Linear-Linear. Zoom the first few hundred Hz by dragging the mouse cursor along the X axis.

d As expected, the spectrum exhibits four sharp peaks at 50 Hz and three harmonics.

20 Return to the Analysis window and select Analyze > Filter. The Electrical Interference section allows us to adjust several parameters. We know that there is no more than the fourth harmonic in our signal. Clampfit determines which harmonics are present and only filters existing ones. So it is mainly a matter of execution speed to restrict the search for harmonics to a realistic number.

a The Cycles to Average setting determines the bandwidth of the filter. Since this parameter is limited by the number of samples in the filtered region, set it to 100.

b Clampfit can automatically determine the line frequency in the signal, but since it is usually known, you should set this parameter manually. Upon OK the filter has completely removed the hum, as you can convince yourself by performing another Power Spectrum and appending it to the existing plot.

c The new power spectrum can be more easily compared if you increase the line thickness of the plots, via the Graph window Properties. In addition, you can compare sweeps 1 and 6 once more.

Not only line frequency hum can contaminate recordings. Other influences such as mechanical vibration may produce broad bands instead of sharp peaks in the power spectrum. Now that you have learned how to produce and to remove a certain type of noise, you might try to introduce wide-band noise and remove it using the Notch Filter. By such practice you will learn a good deal about the benefits and limitations of different filter algorithms. The online Help and Chapter 8, *"Digital Filters"* provide valuable additional information.

If you have episodic data most of the steps we performed to introduce artificial noise can equally be done using Analyze > Arithmetic from the Analysis window. This feature, which requires even less computational effort because no conversion of binary to ASCII data is necessary, is described in the next tutorial. We took a more complicated way to learn about Clampfit's Results window, some of the features it offers, and the input and output of ASCII data. Furthermore, we learned how to use power spectra and digital filtering to improve the signal-to-noise ratio in our recordings.

EVALUATION OF MULTICOMPONENT SIGNALS: SENSILLAR POTENTIALS WITH SUPERIMPOSED ACTION POTENTIALS

In this tutorial we will learn how to use Clampfit for separating the fast and slow components of a signal. This situation occurs especially in extracellular recordings, where contributions of different excitable cells, or different compartments of one cell coincide, forming a multi-component signal.

Open the sample file senspot.dat. Using software filters, we will separate the sensillar, or receptor, potential-the relatively slow downward deflection of the signal-from the much faster action potentials superimposed on it (Thurm 1972, Kaissling & Thorson 1980). Digital filtering requires a single, continuous sampling interval (see Chapter 8, *"Digital Filters"*), which is not the case here with this older data file, as you can see if you select File > Properties. So we have to convert the two sampling intervals of 200 and 600 µs to a single interval using either Analyze > Data Reduction, or Analyze > Interpolation. For comparison we will do both.

Data reduction can be used to eliminate samples from a file acquired at an excess sampling rate, or which has to be reduced in size for other reasons (graphing, minimizing storage requirements, etc.). There are two parameters to set. The reduction factor *n* is the factor by which the number of samples in the file will be reduced. The selection of the reduction method determines how Clampfit eliminates the excess samples.

- **Decimate** retains only every *n*th data point, while the samples in between are deleted. This method minimizes the computational effort, but introduces the danger of aliasing, since the resulting data file is equivalent to one acquired at a lower sampling rate. That means no signal containing higher frequencies than half the sampling frequency in the reduced file may be processed using Decimate reduction.
- **Substitute average**, in contrast, does not introduce any artifacts. Here every n samples are averaged to yield the new data value. This method can eliminate transients and high frequency noise, and can even smooth the signal, since the averaged values are not restricted to the resolution of the A/D converter. The expense for these advantages is the slower performance, which should be relevant only on very slow computers, or with very large data files. Therefore, whenever possible, Substitute average should be the method of choice.
- **Min/Max** determines the largest and the smallest values out of every set of n data points and retains them. Therefore, the minimum for *n* is 2 with this method, which makes it inappropriate for our project. Because, as with Decimate, a subset of the samples in the original file is retained without a change, Min/Max also requires attention to prevent aliasing.

Similar to Data Reduction, Interpolation allows choosing the Interpolation factor and one of two algorithms, Straight line or Cubic spline.

- **Straight line** simply interconnects the existing samples, creating a step-like signal with newly introduced high-frequency components.
- **Cubic spline** generates a smooth signal, again at the price of a slightly slower performance.

In this tutorial, we do not want to increase or decrease the number of samples, but only create a file with a single sampling interval. Therefore, we use the interpolation factor 1:

1 Open the Analyze > Interpolation dialog box.

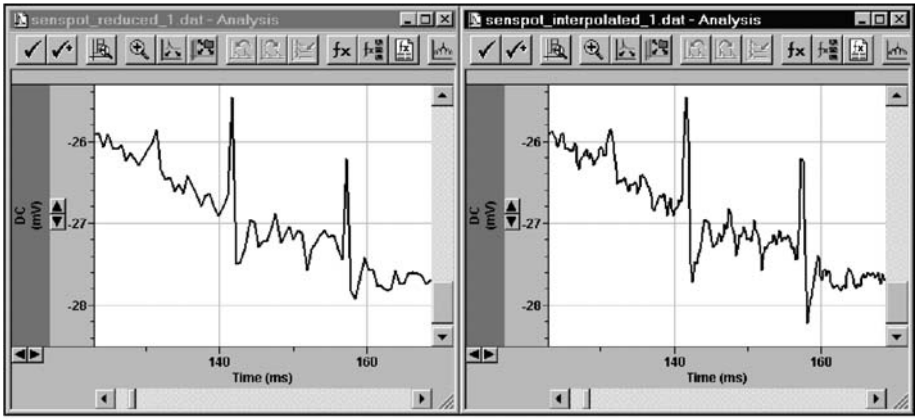

senspot_reduced_1.dat - Analysis
senspot_interpolated_1.dat - Analysis
DC (mV)
-26
-27
-28
140
160
Time (ms)

Figure 7.4: Comparison of original (left) and interpolated (right) files.

2 Select the Cubic Spline method.

3 Select an Interpolation factor of 1.

4 Click OK.

This creates a new file with the higher sampling rate as in the original file, as seen if you have the Analysis windows tiled, and compare the signals after zooming in around 1000 ms. Since we intend to use software filters in the next steps, and the sampling rate determines the range of the allowable cutoff frequency, we will use the interpolated file.

1 Select Edit > Create Duplicate Signal.

2 Make the new signal active by clicking on it, and select Analyze > Filter.

3 First we want to isolate the slow component of the response, that is, the sensillar potential. Several types of lowpass filters are available, which are described in detail in Chapter 8, *"Digital Filters"*.

4 Select individual filters from the Type list to see the cutoff range reported in the Lowpass group change.

5 Select "Gaussian" as the filter type.

6 Specify a -3 dB cutoff frequency of 70 Hz.

7 In the Trace Selection section of the Filter dialog press Select.

8 In the Signals section of the Select Traces dialog and choose The active signal. Since there is only one sweep in the file, we do not need to worry about selecting Traces.

9 Click OK and your selection is reported in the Filter dialog.

10 Since we want to apply the filter to the entire length of the trace, set the Region to filter to Full trace and press OK. The action potentials are removed, except for small residuals.

The signal is called DC2 by default. To give it a more convenient name:

1 Select Edit > Modify Signal Parameters.

2 Enter “70 Hz Lo” in the Name field.

3 Press Save.

How can we be sure now that our choice of the cutoff frequency was appropriate? The filter should have removed the fast action potentials-which is easy to see-without affecting the shape of the slow sensillar potential-which is a little more difficult to determine. The strategy is to subtract the filtered signal from the original one

We can subtract the signals using the Analyze > Arithmetic command. The Arithmetic dialog is very similar to the Column Arithmetic dialog explained in the previous tutorial.

We want to compare the resulting trace to the existing signals in our file. Therefore, we must first create another signal to use as the destination:

1 Select the “70 Hz Lo” signal.

2 Select Edit > Create Duplicate Signal to create the new signal.

3 Open Edit > Modify Signal Parameters.

4 Rename the new signal “Subtracted”.

5 Click Save.

We are now ready to subtract the traces:

1 Open Analyze > Arithmetic.

2 Click the Traces button and select “Subtracted” from the A specified signal list.

3 Click OK.

4 Click the Traces button again and select “DC (mV)” from the A specified signal list.

5 Click OK.

6 From the Operator list select “- (subtract)”.

7 Click the Traces button again and select “70 Hz Lo (mV)” from the A specified signal list.

8 Click OK.

9 The expression you then apply should resemble:

T{VISIBLE} = DC:T{VISIBLE}-70 Hz Lo:T{VISIBLE}.

10 Click OK.

If you like, you can try other filter cutoff frequencies or filter types. The lower the cutoff frequency you specify, the more prominent is the transient before the sensillar potential in the subtracted signal. This indicates that at lower filter frequencies the faster elements of the sensillar potential are more seriously distorted. In contrast, the more you increase the cutoff frequency, the larger are the residuals of the action potentials. The 70 Hz we originally chose was a reasonable compromise.

As you try other filter types, you will notice that the infinite impulse response filters (i.e. all types, but Gaussian and Boxcar) require a comparatively long time to phase in, and additionally start at 0 mV. Chapter 8 in this manual explains these end effects in detail, and in the following paragraphs of this tutorial we will learn two ways to deal with them.

After we have isolated the sensillar potential, we will now take a look at the action potentials. The signal “Subtracted” already fulfills the most important criterion for the evaluation of action potentials: they sit on a non-fluctuating baseline. We will compare that to highpass filtering the signal:

1 Scale the X axis so that the first 500 ms are on display.

2 Select the “DC (mV)” signal.

3 Select Edit > Create Duplicate Signal to create another duplicate of signal DC.

4 Select the new “DC2 (mV)” signal to make it the active signal.

5 Select Analyze > Filter.

6 Select Highpass.

7 Two types of highpass filters are available: a single pole RC filter and an 8-pole Bessel.

8 Select “RC (single pole)” and specify 100 Hz as the -3 db cutoff frequency.

9 Click OK.

10 Select View > Auto Scale > Active Y axis to auto scale the end effect, which starts at a negative value near the original first data point.

11 To remove this transient:

- **a** Bring the cursors into the current region using Tools > Cursors > Bring Cursors.
- **b** Drag cursor 1 to the start of the file and cursor 2 to about 20 ms.
- **c** Select Analyze > Force Values. This function allows you to assign either values that result from the cursor positions, or a fixed value, to a region to force. The Trace Selection again allows specifying the signals and sweeps to include.

d Select Fixed value and enter 0.

e In Region to force, select Cursors 1..2.

f Click OK.

The filtered trace exhibits a considerable residual from the steep portion of the sensillar potential, forming a downward deflection that the earlier action potentials are superimposed on. The flat region of the sensillar potential, where the late action potentials occur, has been effectively removed, however.

The other way of avoiding the end effects is removing the baseline offset before filtering:

1. Select the signal “DC (mV)”.
2. Set cursors 1 and 2 to 0 ms and 50 ms, respectively.
3. Select Analyze > Adjust > Baseline.

This feature corrects an offset or drifting baseline that you can specify using one of four methods. Subtract mean and Subtract slope use the region selected in the drop-down list:

- Mean calculates the average of all data points in the specified region
- Slope fits a regression line, then the result is subtracted from the entire trace, so that the specified region coincides with a straight baseline at y = 0.
- You can enter a fixed value, which is to be subtracted from the selected traces.
- You can correct an irregular baseline if you select Adjust manually and click on the pink correction line to create break points that can be dragged to follow an irregular curve.

Assuming the data values before the onset of the sensillar potential represent the baseline, the most suitable selection for our purpose is the mean between Cursors 1..2:

1. Select Subtract mean of.
2. Select “Cursors 1..2” from the list.
3. Ensure that “Active signal” is selected in the Trace Selection group, and click OK.

As you create duplicates of DC now, they start at zero and the end effects of the filters are minimized. Depending on the application, the amplitude offset in the original data file might be relevant. In this case, you should measure it before you adjust the baseline, or perform the adjustment in the duplicate signal, alternatively.

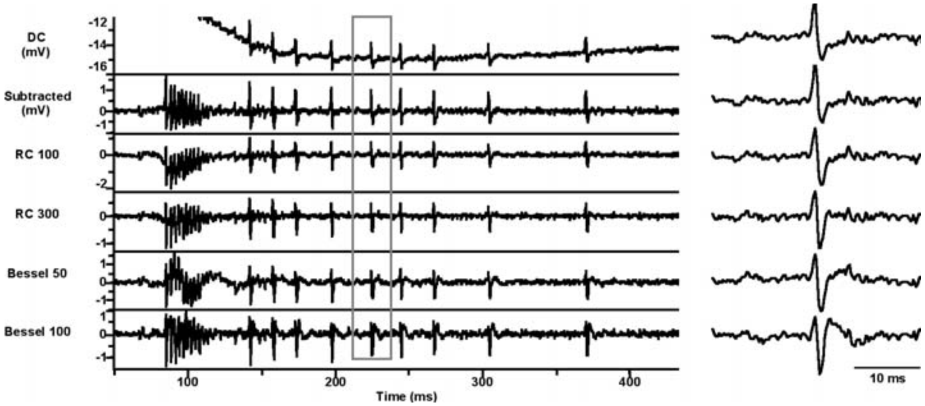


Figure 7.5: Effects of different filter settings.

As you apply a highpass filter to the next duplicate signal now, the end effect does not occur. Try the RC filter at a higher cutoff frequency to see the sensillar potential almost completely eliminated, but the action potential waveform increasingly distorted. At a comparatively low cutoff frequency of 50 Hz, the 8-Pole Bessel highpass filter introduces ringing in response to fast changes in the signal, such as the steep portion of the sensillar potential. If the cutoff frequency is increased to 100 Hz, the ringing disappears, but the action potential waveform is distorted more severely. The right side of the figure above illustrates the influence of the different filter algorithms and frequencies upon the indicated section on the left. Depending on the application, the choice of the suitable highpass filter is far more critical than for lowpass filters. Filtering individual regions of the signal using different algorithms or cutoff frequencies might be necessary for specific applications.

7. Clampfit Tutorials

Now that we have separated the sensillar potential from the action potentials, we can go on evaluating parameters that describe the response. Use Tools > Cursors > Write Cursors to write the times of the action potentials to the Cursors sheet in the Results window or apply Analyze > Statistics. In addition, you can use the cursor measurements to characterize the sensillar potential. Or you can try to mathematically describe the time course of its initial phase:

1 Right-click in the data portion of the “70 Hz Lo” signal and select Maximize Signal from the right mouse menu.

2 Zoom the first 500 ms by clicking and dragging the cursor on the X axis.

3 Set cursors 1 and 2 to 85 ms and 240 ms, respectively.

4 Select Analyze > Fit.

Clampfit provides highly sophisticated fitting features, giving you the opportunity to select from a set of predefined functions, or to specify a custom function. Four different Search Methods are available, in combination with various Minimization and Weighting Methods. A number of additional options are available on the Data/Options tab, and the

Seed Values tab allows you to enter initial parameter estimates, either in a purely numerical way, or with the assistance of a graphical representation. Not all options available in the dialog apply to all datasets, methods, or functions. Since fitting is a fairly complex issue, it is beyond the scope of this tutorial to explain every detail. We strongly recommend that you read Chapter 11, *"Curve Fitting"* for more detailed information on the functions, the algorithms and trouble shooting. In addition, a number of ATF files can be found in the ..\Program Files\Molecular Devices\pCLAMP10.0\Sample Data folder. Their file names indicate their functions and they are ideally suited to help you become more familiar with the Fit feature.

We are now going to investigate whether the steep initial phase of the sensillar potential can be described by an exponential function. Presumably several processes with different time constants superimpose to shape this waveform (Kaissling 1998, Vermeulen & Rospars 1998). Therefore, we will test exponential functions of different orders and compare these models. For this purpose Clampfit provides an automatic comparison of models.

Fitting generally applies to the region between cursors 1 and 2. They already confine the region we are going to fit:

1. Select "Exponential, standard" from the Predefined Function group on the Function/Method tab.
2. Uncheck Select fitting methods automatically and select Levenberg-Marquardt, Sum of squared errors Minimization and no Weighting.
3. On the Data/Options tab, enable Compare Models.
4. The Starting and Ending terms are automatically set to the maximum possible; the 95% Confidence level should be adequate.
5. On the Seed Values tab, you can review or alter the estimates Clampfit has automatically generated. If you change the original values, they can optionally be restored with the Auto-Estimate button.
6. Click OK.

   Clampfit starts fitting exponential functions of increasing order to the data. While the Automatic checkbox in the Compare Models group is enabled, the comparison stops if a higher-order model does not improve the fit. If the Ending terms are set manually, however, Clampfit continues up to the specified order.
7. Select Analyze > Fitting Results. The Parameters and the fit Statistics are displayed on two tabs. The Model number spinner in the bottom of the dialog allows you to review the values for the different models. In the Best Model group on the Statistics tab Clampfit reports that, under the specified fitting conditions, a second order exponential function reasonably describes the fitted portion of the data.
8. Use the Clipboard or Lab Book buttons to copy to the respective destinations the results for the model that is currently displayed in the dialog.

SEPARATING ACTION POTENTIALS BY THEIR SHAPE

Electrophysiological recordings often comprise rather similar signals, which slightly differ in their amplitudes or kinetics because they originate from different cells. One well-known example of these signals are action potentials extracellularly recorded from insect sensilla (Frazier & Hanson 1986, Schnuch & Hansen 1990). This tutorial will demonstrate how Clampfit can be used for characterizing the waveforms in a series of similar signals. Then we will investigate whether the differences in the measured parameters permit us to assign them to different subpopulations, and finally we will save them in two separate files for further evaluation.

1 Open the sample file spikes02.abf. The file contains spontaneous action potentials of two different types, which were extracellularly recorded from an insect olfactory sensillum. The file was acquired in high-speed oscilloscope mode, using the highpass-filtered signal AC as the trigger channel. For waveform analysis, only an unfiltered signal is relevant:

2 Right-click in the data area of the signal “DC” and select Maximize Signal.

3 One of the sweeps was obviously triggered by an artifact. You can exclude it from the analysis by deselecting it and performing all analyses on the Visible Traces only.

4 Click on the artifact to make it the active sweep. In the lower right corner of the Analysis window Clampfit reports the number of the active sweep as 63.

5 Open View > Select Sweeps

6 Select sweep 63 and press the Invert button.

7 Click OK.

8 In sweep 39 there is another artifact. You can either remove it using Analyze > Force Values, or exclude it from the analysis by specifying the Region to analyze with cursors 1 and 2. We will use the second method here.

There is a slow drift in the baseline the action potentials sit on, as can be seen best in View > Data Display > Concatenated mode. Depending on the recording method, slow drifts can have different origins. Frequently seen is a slowly drifting electrode potential, when non-chlorided metal electrodes such as tungsten or platinum are used (Geddes 1972). Insect sensilla often exhibit slow, occasionally oscillatory, changes in their steady state potential, whose origin is not completely understood. A common way to deal with slowly drifting signals are AC-coupled recordings. In the previous tutorial, possible effects of highpass filtering on the signal waveform were demonstrated for digital filters. Comparing the signals AC and DC in the sample file, we can see that analog filters equally affect the waveform. Therefore, we remove the drift in the baseline using Analyze > Adjust > Baseline:

1 First, use Edit > Create Duplicate Signal. We are going to use this to compare the Subtract mean and Adjust manually baseline adjustment methods.

2 Right-click in the data area and select Show Acquisition Signals to show all signals.

3 Select the signal “DC”.

4 Select View > Data Display > Sweeps.

5 Set cursors 1 and 2 to 0 and 2 ms, respectively. If the X axis units are in seconds you can change to milliseconds using the Units list on the X Axis tab of the View > Window Properties dialog box.

6 Select Analyze > Adjust > Baseline.

7 Select Subtract mean of Cursors 1..2. Ensure the Trace Selection is “Active signal” and “All visible traces”.

8 Click OK.

9 Select View > Data Display > Concatenated.

10 Select the signal “DC2” and select Maximize Signal from the right mouse menu.

11 Open the Analyze > Adjust > Baseline dialog and select Adjust manually. Click OK.

12 A pink correction line is displayed, which can be dragged to shape by clicking on it to create moveable break points. The X and Y coordinates of the most recently altered point are reported in a floating toolbar. When the line closely follows the drift of the baseline, press OK

13 Select View > Data Display > Sweeps and Show Acquisition Signals to inspect the differences between the two methods.

While the manual adjustment exhibits a surprising accuracy and can lead to a satisfying result depending on the application, in this case Mean adjustment results in less variation between sweeps. So we will use the signal “DC”.

The next step in the waveform analysis is aligning the peaks of the action potentials. In the original file the signals are aligned at the threshold crossing point at 4 ms (100 samples). Clampfit can shift sweep type data to either direction for a fixed time interval, which can be entered numerically, or alternatively defined by the position of one of the cursor pairs.

Analyze > Time Shift removes a number of samples corresponding to the shifted interval at one end of the Analysis window. There are two options to deal with these “wrapped” samples. Rotate samples adds them to the other end of the sweeps. This option is useful when you are not completely sure about the Time Interval, because a Time Shift can be undone by shifting in the opposite direction. However, the rotated data points are not separated from the rest of the data in any way. So you should be aware that every time shift affects the integrity of the data. The other option, Replace wrapped samples with zeros, makes it easier to recognize the samples that are not part of the actual data file. The time shift cannot be undone, however, unless the original file is reopened.

The waveform analysis we are going to do requires that all action potentials be aligned with a prominent element, namely their positive peaks:

1 Move cursors 1 and 2 so that they encompass the peaks but exclude the artifact in sweep 39.
2 Open Analyze > Time Shift.
3 Select Align Peaks.
4 Select “DC (mV)” from the Signal to search list.
5 Select “Cursors 1..2” from the Region to search list.
6 Since the interval to shift is different for each individual sweep, which excludes undoing the shift anyway, select Replace wrapped samples with zeroes.
7 Ensure that All visible traces is set for the Trace Selection.
8 Click OK.

Next, using Analyze > Statistics we are going to determine waveform parameters that might be characteristic for the individual types of the action potentials. The measurements Clampfit can perform on the data file are essentially the same that Clampex offers during acquisition, so if the recordings are well-reproducible regarding time course and amplitude, it will save time if you set up Clampex to perform the measurements in real-time.

1 Set the cursors 1 and 2 to 5 ms and 10.8 ms, respectively, so they include the action potentials at their bases, but exclude the transient in sweep 39.
2 Ensure that “DC (mV)” is selected in the Trace Selection group, and that Peak Polarity is set to Positive-going.
3 Ensure that the Search Region is set to “Cursors 1..2”.
4 For our action potentials, not all Clampfit measurements are relevant. While the Peak amplitude is interesting, the Time of peak is identical for all sweeps, because we aligned the peaks. The Antipeak amplitude and the Time of antipeak might be different for the two types of spikes and can be further evaluated. The other measurements, such as the Area under the peaks and the kinetics of rise and decay, as well as the Half width might yield differences between the two spike types and should be included in the further evaluation. Select all relevant measurements.
5 Upon OK Clampfit writes the measurements to the Statistics sheet in the Results window.

We assume that the action potentials belong to two populations. First we will distribute them into two groups, evaluating their peak-to-peak amplitude. Then we will investigate whether the other parameters in these two groups are significantly different from each other.

In the Results window Statistics tab select the column named “R1S1 Time of Antipeak (ms)” by clicking on its header. Now select Edit > Insert > Columns to create a new column. Double-click on the new column’s title and rename it “Peak-Peak”.

For computing the peak-to-peak amplitude, use Analyze > Column Arithmetic to subtract the "Antipeak" from the "Peak" column (see the second tutorial for details on Column Arithmetic). The peak-to-peak time can be calculated by subtracting 6 ms (i.e. the time of the positive peaks after aligning) from the "Antipeak (ms)" column.

There are different ways of determining the threshold between the large and the small action potentials. The simplest way is by creating a scatter plot. If there are clear differences, they should show up with this comparatively coarse method:

1. Select the "Peak-Peak" column on the Statistics sheet by clicking on the column header, and go to Analyze > Create Graph. By default, the selection is assumed to include the dependent variable Y, and is plotted versus an incrementing X.
2. Right-click on the plot and select Properties, or alternatively go to View > Window Properties in the main menu to open the Window Properties dialog.
3. On the Plots tab set the Curve Type to Scatter and click OK to remove the connecting lines.

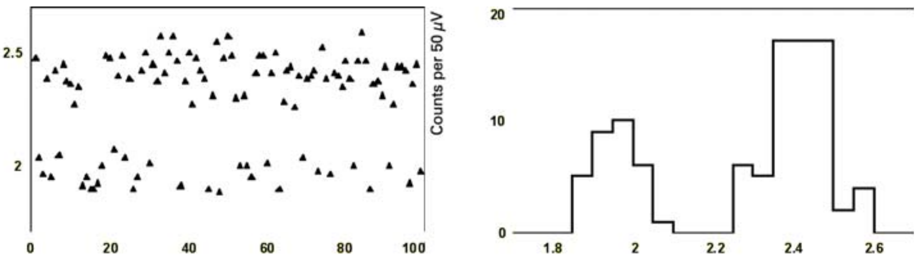


Figure 7.6: Scatter plot and histogram showing groupings in peak-peak measurements.

In our example, there are obviously no peak-to-peak amplitudes between 2.10 and 2.25 mV. Another way to determine the threshold, which can also reveal less evident differences, is a frequency distribution histogram. This requires the definition of bins.

1. With the "Peak-Peak" column selected, go to Analyze > Basic Statistics. Of the Statistics Options that are available in this dialog box, only the Number per category is relevant for our project.
2. Select Perform Breakdown Analysis, select "Peak-Peak" in the Category column drop-down, check Bin the categories, and press Specify Bins. A dialog is displayed that allows you to specify bins of user-defined or fixed size. To cover the entire range of the occurring peak-to-peak amplitudes, specify 20 Fixed bins with the Initial Value 1.7 mV and a Width of 0.05 mV.
3. Click OK in the Specify Bins dialog and OK in the Basic Statistics dialog. Clampfit writes the statistics on the Basic Stats sheet in the Results Window.

4 Select the “Bin center” column first and press the X toolbutton to make it the X column, which is reported in the column header. Then select the “#/Cat” column and press the Y+ button.

5 Now go to Analyze > Create Graph, or press the toolbutton. Unless a different Graph Window is open, the default template is a Line Graph. Select Properties from the right-click menu and, on the Plot tab, set the Plot Type to Histogram. On the Y Axis tab, enter 0 in the Bottom limit field.

After clicking OK, the plot should look similar to the histogram in Figure 7.6. Again there is no amplitude around 2.15 mV. So we can state the null hypothesis that the action potentials with a peak-to-peak amplitude below this threshold are different from those having an amplitude above. In the following steps, we will test this hypothesis.

Clampfit features a number of statistical tests, which are accessible in the bottom section of the Analyze menu. See the online Help for details on their use and their algorithms. For each of the parameters we determined, we want to compare two samples of unequal size, and calculate the probability that they originate from two different parent populations. Several tests can be used to investigate this question: an F-Test followed by an unpaired Student’s t-Test, a nonparametric Mann-Whitney U-Test, and One-Way ANOVA (Sokal & Rohlf 1981). The general approach is identical for all statistics features available from the Results window, so only the F- and t-Tests are demonstrated here.

1 On the Statistics tab of the Results window select the columns “R1S1 Peak Amp” through “R1S1 Half-width (ms)” and go to Analyze > F-Test and Student’s t-Test.

2 Check F-Test and the two Unpaired options in the Student’s t-Test group. The lower half of the dialog is identical to the Basic statistics dialog. The Column selection should still be All selected columns, Perform Breakdown Analysis with Bin the categories from the column “Peak-Peak” still active.

3 In contrast to the previous procedure, this time we use two user-defined bins, comprising the large and the small peak-to-peak amplitudes, respectively. Click Specify Bins and select Defined Bin Size. Using the Add button, specify two bins, the first ranging from 0 to 2.15 mV, and the second from 2.15 to 5. The Edit and Delete buttons are only enabled when one of the bins in the Bin Number column is highlighted.

Upon OK, the Lab Book window is displayed, reporting a number of parameters for every column we included in the analysis. At the beginning of each column-related section, general information about the input values is reported, such as the sample size, the mean, and the variance within each bin. Then the test results are reported, which in our case includes the F-value and F-probability. This parameter indicates whether the variance of the two groups is different. Below that, the results of the t-test are listed: the t-value, the probability, and the degree of freedom.

Depending on the F-probability, use the t-probability from the pooled or the separate t-test in the further course, even if you find that there is virtually no difference for our example. A probability value of 0.0000 means that the probability that all peak amplitudes of the action potentials belong to the same population is less than 10^{-4}.

Table 7.1: Statistical results.

============ H: Peak	========= 0<=Bin 0<2.15	========= 2.15<=Bin 1<5	
Sample Size	31	68	
Mean	1.3132	1.7075	
Variance	0.0041	0.0052	
Degree of freedom ============	30 =========	67 =========	
F-Value	1.2461		
Probability	0.5129		
	t-Value	Probability (2-tail)	Degree of freedom
============ Unpaired (pooled var.)	========= -26.1606	========= 0.0000	========= 97
Unpaired (separate var.) ============	-27.2656 =========	0.0000 =========	64 =========

To summarize, of the 12 parameters we evaluated, 5 are significantly different for the two types of action potentials we distinguished ($p < 10^{-4}$). They are all correlated with the amplitude. The peak-to-peak amplitude is not included, since it was the basis of our null hypothesis. For the reasons explained above, Time of peak, Mean and Standard deviation are also not considered here. Virtually all parameters that describe the time course of the action potentials are not significantly different for the two groups ($p > 0.05$). Only the time of the greatest right slope ($p = 0.04$) is different, but on an extremely weak basis. So we have collected good evidence that the data file contains two types of action potentials with different amplitudes. For further evaluation we are going to save them in separate files now.

First we sort them in the order of increasing peak-to-peak amplitude:

1 In the Results window, select the “Peak-Peak” column and go to Analyze > Sort. The current selection determines the default Key Column, but you can also highlight any other column in the drop-down list.

2 The default Row Range also depends on the selection, in our example it includes all non-empty rows. The Columns To Sort should normally comprise all columns, because the relation between individual columns gets lost if only a subset of them is selected here. The data can be sorted in either Ascending or Descending Sort Direction, and finally sorting can be performed in a Case sensitive way, if the Key Column contains strings.

3 Click OK.

4 After sorting, scroll down until you reach the gap in between the sub- and suprathreshold peak-to-peak amplitudes. The last subthreshold value should be found in Row 31. To better separate the two result blocks, select row 32 by clicking on its header and go to Edit > Insert > Rows.

5 Select the subthreshold Rows in the “Trace” column and sort them in ascending order:

   a Tile the Results and Analysis windows vertically, select the Analysis window and go to View > Select Sweeps

   b In the By Selection field, highlight those sweeps whose numbers are in the Trace column now, all the time holding down the <Ctrl> key.

   c Before you press OK, select By user-entered list and copy the contents of the Range field, which exactly reflects your selection, to the Windows clipboard. Depending on the application window size, you might have to select the sweeps in two portions, scrolling down in the Results window. This can easily be done, if you again press <Ctrl> before you start highlighting the remainder of the sweeps during the second turn. Upon OK, only the small action potentials are on display.

6 Save the file as spikes_s.abf now, making sure that the Data selection reported in the Save Data As dialog is Visible sweeps and signals.

7 Reopen the original data file, adjust the baseline, align the peaks and open the select sweeps dialog once more.

8 Paste the clipboard contents into the Range field and press the Invert button. the selection now includes the sweep with the artifact (#63), so deselect it by clicking while you hold down <Ctrl>. Then save the Visible sweeps as spikes_l.abf for further evaluation.

8. Digital Filters

In digital signal processing a *system* is something that operates on one or more *inputs* to produce one or more *outputs*. A *digital filter* is defined as a system (in the case of Clampfit, a software algorithm) that operates on digitized data to either pass or reject a defined frequency range. The objective of digital filtering is to remove undesirable frequency components from a digitized signal with minimal distortion of the components of interest.

There will be instances when it is necessary to filter experimental data after they have been digitized. For example, you might want to remove random noise or line frequency interference from the signal of interest. To this end, Clampfit offers several types of digital filters.

The lowpass filters include Bessel (8-pole), boxcar, Butterworth (8-pole), Chebyshev (8-pole), Gaussian, a single-pole RC and an 8-coincident-pole RC. The highpass filters include Bessel (8-pole) and 8-coincident-pole RC. The Gaussian and boxcar filters are finite impulse response (FIR) filters while the Bessel, Butterworth, Chebyshev and RC filters are infinite impulse response (IIR) filters (see following section: *"Finite vs. Infinite Impulse Response Filters"*).

A notch filter is available to reject a narrow band of frequencies and an electrical interference filter is provided to reject 50 or 60 Hz line frequencies and their harmonics.

FINITE VS. INFINITE IMPULSE RESPONSE FILTERS

Digital filters can be broadly grouped into finite impulse response (FIR) filters and infinite impulse response (IIR) filters. FIR filters are also referred to as *nonrecursive* filters while IIR filters are referred to as *recursive* filters.

The output of FIR filters depends only on the present and previous inputs. The general "recurrence formula" for an FIR filter, which is used repeatedly to find successive values of *y*, is given by:

$$y_n = \sum_{k=0}^{M} b_k x_{n-k}$$

where y_n is the output value for the n^{th} point x and b_k is the k^{th} of M filter coefficients. In the case of the Gaussian and boxcar filters in Clampfit, the M points ahead of the current point are also used, giving a general recurrence formula of:

$$y_n = \sum_{k=-M}^{M} b_k x_{n-k}$$

where the *filter width* is $2(M + 1)$ points.

The disadvantage of FIR filters is that they can be computationally inefficient as they might require several tens, hundreds or even thousands of coefficients depending on the filter characteristics.

The advantages are that FIR filters are inherently stable because there is no feedback and they possess ideal linear phase characteristics, exhibiting no phase distortion. That is, all frequency components passing through the filter are subject to the same pure time delay.

On the other hand, the output of IIR filters depends on one or more of the previous output values as well as on the input values. That is, unlike FIR filters, IIR filters involve feedback. The general recurrence formula for an IIR filter is given by:

$$y_n = \sum_{j=1}^{N} a_j y_{n-j} + \sum_{k=0}^{M} b_k x_{n-k}$$

where a and b are the N and M filter coefficients, where a represents the feedback coefficients. Note that the value of y for a given point n depends on the values of previous outputs y_{n-1} to y_{n-N} as well as the input values x.

The major advantage of IIR filters is that they are computationally more efficient, and therefore much faster, than FIR filters. The disadvantages are that IIR filters can become unstable if the feedback coefficients are unsuitable, and recursive filters cannot achieve the linear phase response that is characteristic of FIR filters. Therefore, all IIR filters introduce a phase delay to the filtered data.

The problem of potential instability of IIR filters is solved in Clampfit by limiting the cutoff frequencies for all filter types to a range where the response is always be stable (see *"Cutoff Frequency Limitations"* on page 154). However, the phase delay is not corrected.

The Nyquist rate (see *"The Sampling Theorem in Clampfit"* on page 15) has important consequences for digital filtering in that the maximum analog frequency that a digital system can represent is given by:

$$f_h = \frac{1}{2T}$$

where T is the minimum sampling interval and f_h is the Nyquist frequency.

As a consequence of this, the maximum filter cutoff frequency of digital filters is limited to one-half the sampling rate. That is, the ratio of the cutoff frequency to the sampling rate (f_C/f_S) cannot exceed 0.5. In fact, only the Gaussian and single pole RC filters can realize an f_C/f_S ratio as high as 0.5; the Bessel, Butterworth, Chebyshev and notch IIR filters are limited to values that are somewhat lower than this because of performance degradation at higher f_C/f_S ratios (see *"Cutoff Frequency Limitations"* on page 154).

The f_C/f_S ratio limitation should not present a problem if antialias filtering and oversampling are judiciously applied. For example, with a lowpass antialiasing filter cutoff of 4 kHz and a sampling rate of 40 kHz, an f_C/f_S ratio limited to as low as 0.1 allows a maximum cutoff frequency of 4 kHz, which is well above any useful cutoff frequency that might be applied to this particular digitized record.

DIGITAL FILTER CHARACTERISTICS

An ideal filter would have a rectangular magnitude response with no attenuation in the passband and full attenuation in the stopband. However, ideal filters are *noncausal* in that the present output depends on future values of the input. They are, therefore, not realizable. However, realizable digital filters can approximate ideal filters in that the output can be delayed for a finite interval until all of the required inputs have entered the system and become available for determination of the output.

Different filter types optimize different characteristics of the ideal filter that they are meant to approximate. Therefore, the application of a particular filter type should be carefully considered in view of the specific requirements at hand.

Most filters introduce a time lag between the input and output signals. Depending on the filter type, some frequencies are subjected to a greater lag than others. As a consequence the output signal is distorted to some degree. This distortion takes the form of "ringing" and "overshoot" in the filter output given a step function input (e.g. a square pulse). Filters that introduce equal time lags for all frequencies are said to have a constant "group delay". Such filters exhibit minimal ringing and overshoot.

A filter can be characterized by its cutoff frequency and steepness of response. In the Clampfit filters the cutoff frequency is defined as the frequency at which the signal amplitude decreases by a factor of 2. This corresponds to a drop in power of $1/\sqrt{2}$, or -3 decibels (dB).

The steepness of a filter, its "roll off," defines the rate at which a signal is attenuated beyond the cutoff frequency. It is desirable to have as steep a roll off as possible so that unwanted frequencies are maximally attenuated. However, filters that are designed for maximally steep rolloffs necessarily sacrifice constant group delay characteristics, and therefore exhibit ringing and overshoot.

The steepness of a filter response is also a function of its order (number of poles): the higher the filter order, the steeper the response. Apart from the single pole RC filter, the IIR filters in Clampfit are 8-pole realizations.

END EFFECTS

All software filters exhibit “end effects” at the beginning or end of a data record. In the case of the boxcar and Gaussian filters, end effects occur at both ends of the record because these filters use both previous and succeeding points to filter the current point. Clearly, at the beginning of the record only succeeding points are available. These filters are, therefore, phased in progressively as previous points become available. Towards the end of the record fewer and fewer following points become available so the filter is progressively phased out. The filter coefficients are adjusted during these phases in accordance with the available number of points.

Filters with fewer coefficients exhibit shorter end effects as the full operating width overlaps a fewer number of points. The number of settling points required for the Gaussian and boxcar filters is (filter length –1)/2. In the case of the Gaussian the filter length is equal to the number of coefficients, while in the case of the boxcar the filter length is equal to the number of averaging points.

IIR filters exhibit startup-transients only. Since these filters do not use points ahead of the current point for the filter output there is no phase-out transient. The output of these filters depends on previous outputs as well as the current input. Therefore, a certain number of points must be processed before these filters reach stability.

The rise time (T_r) of a lowpass filter can be estimated from $T_r = 0.35/f_c$. As the filter settles to within about 10% of its final value in one rise time, a duration of $3 \times T_r$ is sufficient to allow the filter to reach 0.1% of its final value, requiring about $(3 \times 0.35/f_c) \times f_S$ points.

BESSEL LOWPASS FILTER (8 POLE) SPECIFICATIONS

- 10% to 90% step rise time: $0.3396/f_c$
- Maximum overshoot: 0.4%
- Attenuation: 114 dB at $f = 10\ f_c$

A Bessel lowpass filter has a maximally flat response over the entire frequency range (constant group delay characteristics), exhibiting minimal overshoot and ringing in response to a step function. As all frequencies are delayed equally the shape of the original signal is preserved. Because of these characteristics, Bessel filters are most commonly used for time-domain analysis of biological data, where the preservation of the shape of the original signal is critical.

Expected vs. Observed Overshoot

100 mV step pulse, f_S = 10 kHz

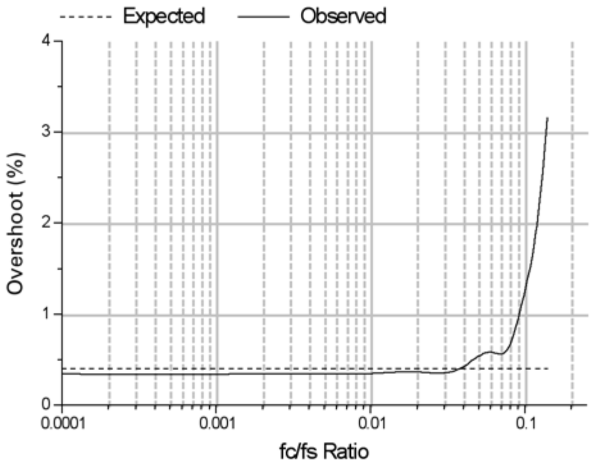


Figure 8.1: Bessel lowpass filter (8 pole) expected vs. observed overshoot.

Expected vs. Observed Rise Times

100 mV step pulse, f_S = 10 kHz

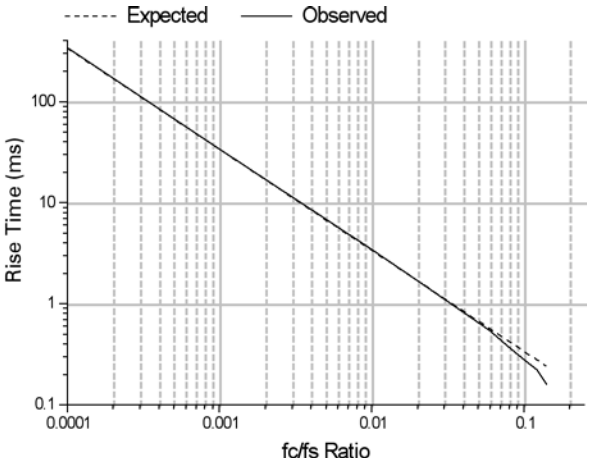


Figure 8.2: Bessel lowpass filter (8 pole) expected vs. observed rise time.

Normalized Frequency Response

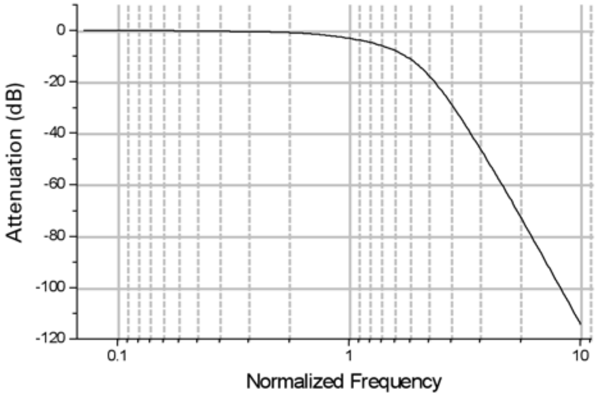


Figure 8.3: Bessel lowpass filter (8 pole) normalized frequency response.

BOXCAR SMOOTHING FILTER SPECIFICATIONS

Smoothing filters are generally used to remove high-frequency components from slowly varying signals and are therefore lowpass filters. The boxcar-smoothing filter uses the average of the current point and a given number of previous and succeeding points to replace the value of the current point. The recurrence formula for a boxcar filter is:

$$y_n = \sum_{k=-M}^{M} \frac{X_{n-k}}{P}$$

where x_n is the nth point to be filtered (at $k = 0$), P is the number of smoothing points (the filter width) and $M = (P - 1)/2$. The boxcar filter does not introduce a time lag.

Like the other filters, the boxcar filter also attenuates the signal; the degree of attenuation is directly proportional to the frequency of the signal and the number of smoothing points. The Figure 8.4 compares the attenuation of 10, 50 and 100 and 500 Hz sine waves (sampled at 10 kHz) at various filter lengths:

Boxcar Filter Attenuation vs. Number of Smoothing Points

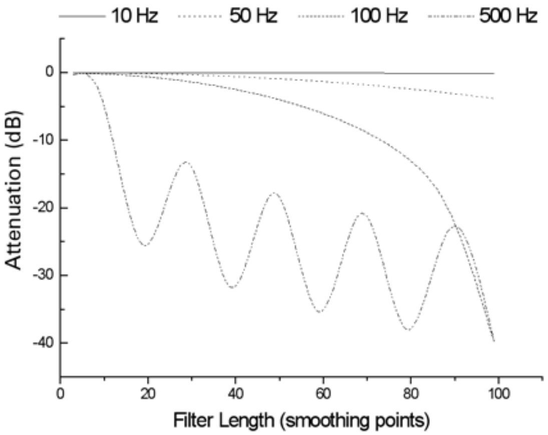


Figure 8.4: Boxcar filter attenuation vs. number of smoothing points.

Note that filtering periodic signals with the boxcar filter can introduce a periodic attenuation response, as seen with the 500 Hz signal. This occurs because the filter output for the current point is the mean of its value and the values of its immediate neighbors. The output therefore depends on the relative proportion of high and low data values within a given filter length.

BUTTERWORTH LOWPASS FILTER (8 POLE) SPECIFICATIONS

- 10% to 90% step rise time: $0.46/f_c$
- Maximum overshoot: 16.0%
- Attenuation: 160 dB at $f = 10\ f_c$

The Butterworth lowpass filter has a maximally flat response at low frequencies and a monotonically decreasing amplitude response with increasing frequency. The group delay is not constant so the Butterworth filter exhibits ringing and a substantial overshoot in response to a step function. This filter, however, has sharper roll-off characteristics than the Bessel filter. It is, therefore, better suited than the Bessel for frequency domain applications such as noise analysis. However, because of its nonconstant group delay characteristics this filter should generally not be used for time-domain analysis of biological data.

Expected vs. Observed Overshoot

100 mV step pulse, f_S = 10 kHz

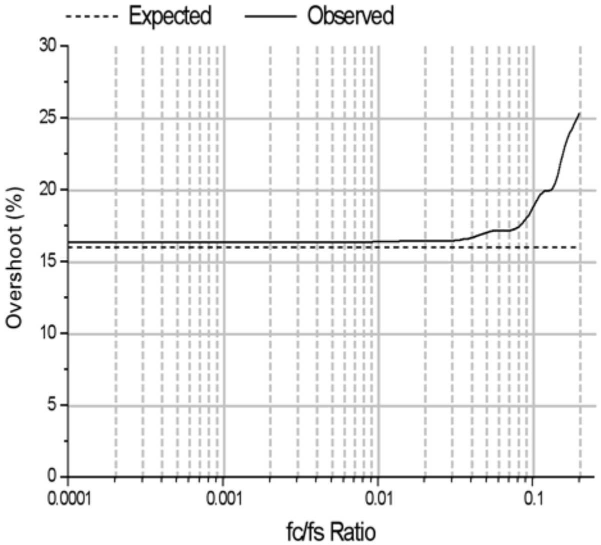


Figure 8.5: Butterworth lowpass filter (8 pole) expected vs. observed overshoot.

Expected vs. Observed Rise Times

100 mV step pulse, f_S = 10 kHz

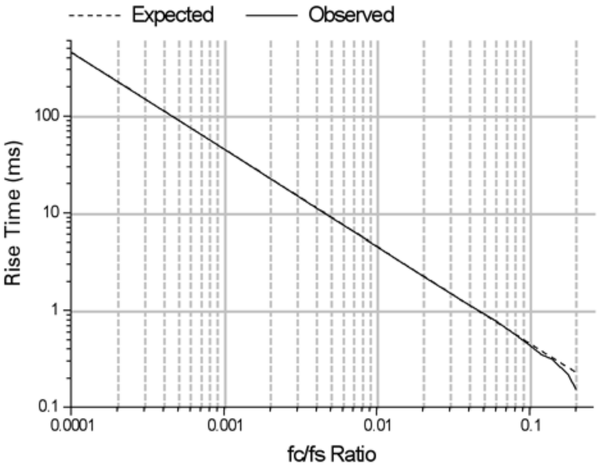


Figure 8.6: Butterworth lowpass filter (8 pole) expected vs. observed rise time.

Normalized Frequency Response

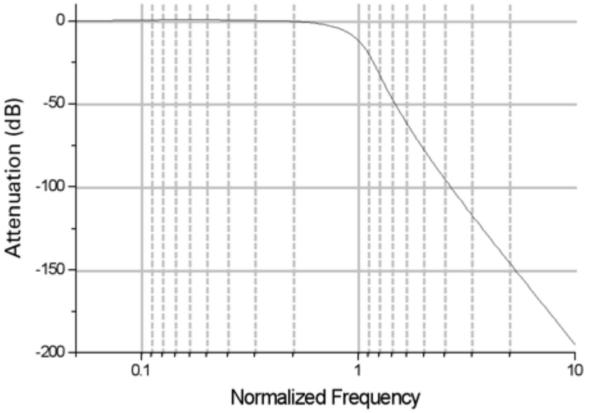


Figure 8.7: Butterworth lowpass filter (8 pole) normalized frequency response.

CHEBYSHEV LOWPASS FILTER (8 POLE) SPECIFICATIONS

- 10% to 90% step rise time: 0.53/f_c
- Maximum overshoot: 16.0%
- Attenuation: 193 dB at f = 10 f_c

The Chebyshev lowpass filter has a maximally sharp transition from the passband to the stopband. This sharp transition is accomplished at the expense of ripples that are introduced into the response. The Chebyshev filter in Clampfit has a fixed ripple of 1 dB. Like the Butterworth, the sharp roll-off characteristics of the Chebyshev filter make it suitable for analysis of data in the frequency domain, such as noise analysis. Although the Chebyshev filter has a sharper roll-off than the Butterworth, it exhibits an even larger overshoot and more ringing. Therefore, it is also not generally suitable for time-domain analysis of biological data.

Expected vs. Observed Overshoot

100 mV step pulse, f_S = 10 kHz

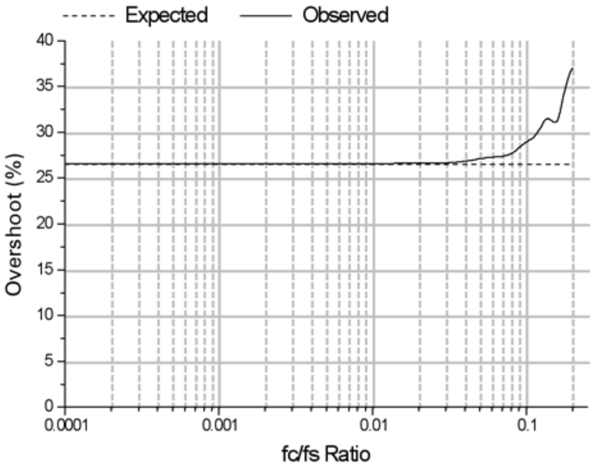


Figure 8.8: Chebyshev lowpass filter (8 pole) expected vs. observed overshoot.

Expected vs. Observed Rise Times

100 mV step pulse, f_S = 10 kHz

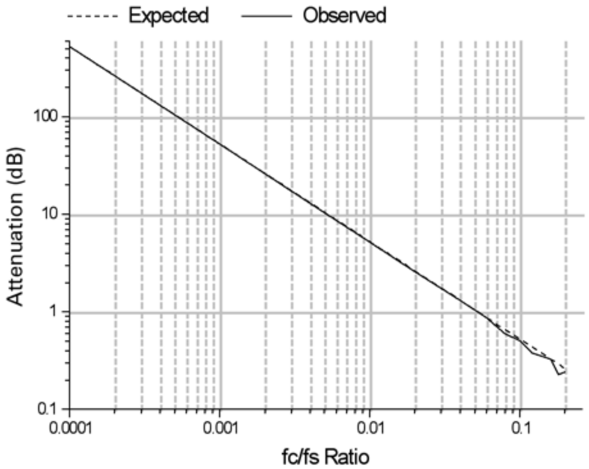


Figure 8.9: Chebyshev lowpass filter (8 pole) expected vs. observed rise time.

Normalized Frequency Response

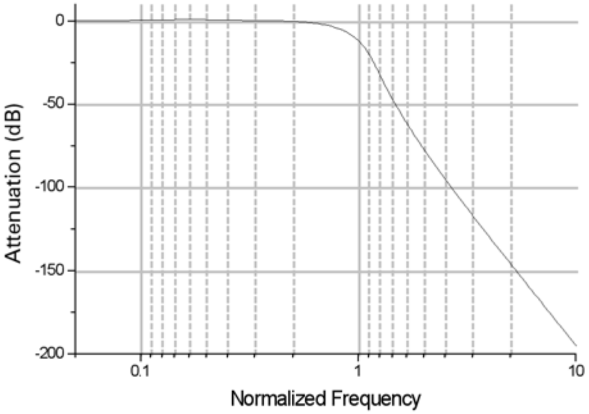


Figure 8.10: Chebyshev lowpass filter (8 pole) normalized frequency response.

GAUSSIAN LOWPASS FILTER SPECIFICATIONS

- 10% to 90% step rise time: 0.3396/f_c
- Maximum overshoot: 0%
- Attenuation: 90 dB at $f = 10\ f_c$

The Gaussian lowpass filter forms a weighted sum of the input values to form an output value according to the following recurrence formula:

$$y_i = \sum_{j=-n}^{n} a_j x_{i-j}$$

where a_j are the Gaussian coefficients that sum to unity. The algorithm for and properties of this filter are thoroughly described by D. Colquhoun and F.J. Sigworth (1995).

The Gaussian filter is particularly suited for filtering biological data for analysis in the time domain as it produces no overshoot or ringing and introduces no phase delay. The disadvantage is that it can be slow at high f_c/f_s ratios where the number of filter coefficients is large.

Expected vs. Observed Rise Times

100 mV step pulse, f_S = 10 kHz

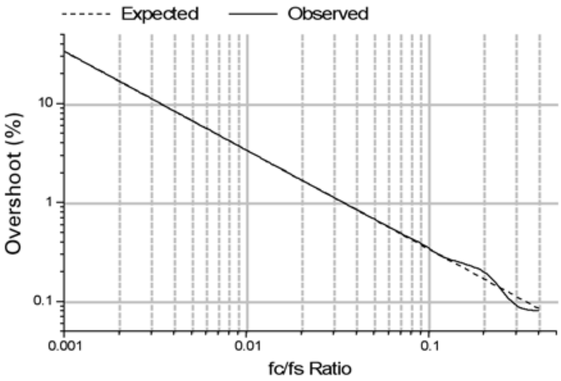


Figure 8.11: Gaussian lowpass filter expected vs. observed rise time.

Normalized Frequency Response

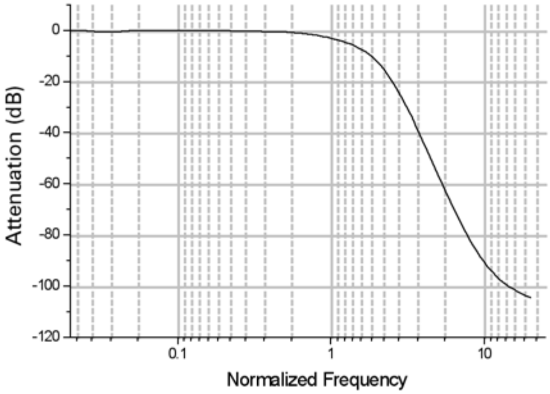


Figure 8.12: Gaussian lowpass filter normalized frequency response.

NOTCH FILTER (2 POLE) SPECIFICATIONS

The number of poles of the notch filter is fixed at two. This filter has essentially zero gain (–inf dB) at its center frequency and about unity gain (0 dB) elsewhere. The notch filter has approximately zero phase shift except at its center frequency, at which the phase shift is undefined (because the gain is zero). In both respects (magnitude and phase) the resonator behaves like a "real" analog tuned circuit.

Figure 8.13 shows the frequency response of the notch filter with a 60 Hz center frequency with a 10 Hz –3 dB width.

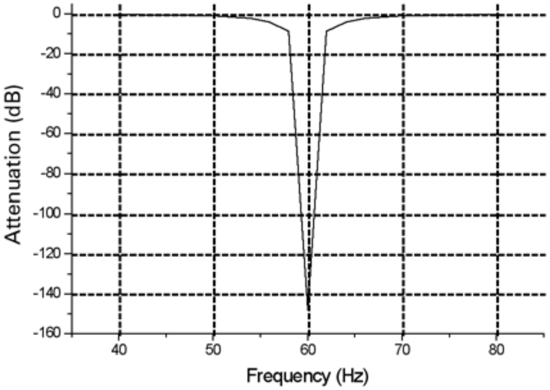


Figure 8.13: Notch filter (2 pole) frequency response (60 Hz center frequency, 10 Hz –3 dB width)

Settling Points

The –3 dB width has a significant influence on the number of points required for the notch filter to settle, the narrower the notch the greater the number of settling points. For example, for a 60 Hz sine wave sampled at a frequency of 1 kHz, applying a 60 Hz notch filter with a 1 Hz –3 dB width requires 2000 sampling points for the filter to reduce the amplitude to 0.1% of its original value (–60 dB). In contrast, a 60 Hz notch filter with a 10 Hz –3 dB width requires only 200 points.

The number of settling points also increases with increasing sampling frequency. For example, a notch filter with a –3 dB width of 10 Hz requires 2000 settling points for data sampled at 10 kHz compared to 200 for data sampled at 1 kHz.

The relationship between the number of settling points (P_s) and the sampling frequency (f_s) and –3 dB notch width (W_{-3dB}) is given by:

$$P_s = \frac{2f_s}{W_{-3dB}}$$

for attenuation of the center frequency by 60 dB.

RC LOWPASS FILTER (SINGLE POLE) SPECIFICATIONS

- 10% to 90% step rise time: $0.3501/f_c$
- Maximum overshoot: 0%
- Attenuation: 20 dB at $f = 10\ f_c$

The RC lowpass filter function is equivalent to that of a simple first-order electrical filter made up of a single resistor and a single capacitor. The RC filter introduces a phase delay to the output, but the group delay is constant so that there is no ringing or overshoot.

The recurrence formula for this filter is:

$$Y(n) = X(n) + W + (Y(n-1) - X(n-1))$$

where $Y(n)$ is the current output, $X(n)$ is the current data point, $Y(n–1)$ is the output for the previous point and:

$$W = e^{-dt/\tau} \qquad \text{where} \qquad \tau = 1/2\pi f$$

where dt is the sampling interval and f is the –3 dB cutoff frequency.

Expected vs. Observed Rise Times

100 mV step pulse, f_s = 10 kHz

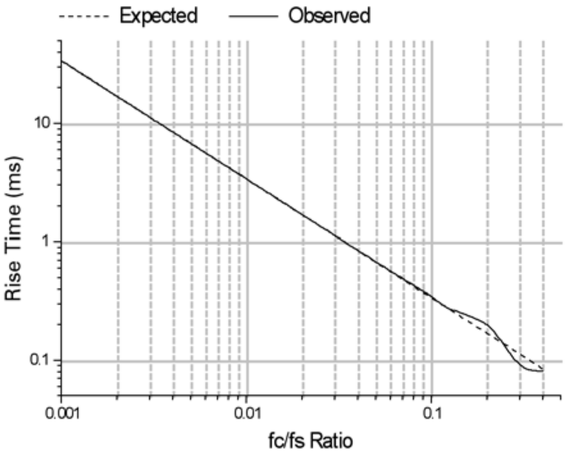


Figure 8.14: RC lowpass filter (single pole) expected vs. observed rise time.

Normalized Frequency Response

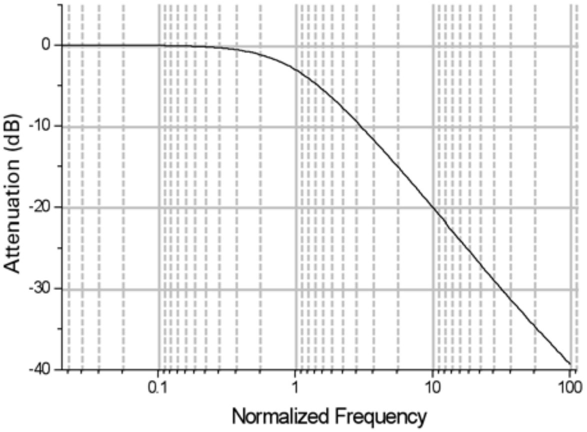


Figure 8.15: RC lowpass filter (single pole) normalized frequency response.

RC LOWPASS FILTER (8 POLE) SPECIFICATIONS

- 10% to 90% step rise time: $0.34/f_c$
- Maximum overshoot: 0%
- Attenuation: 80 dB at $f = 10\ f_c$

The 8-pole RC filter is a “multiple coincident pole” realization where the data points are filtered by applying 8 single pole RC sections in succession. The recurrence formula for this filter is, therefore, identical to that of the single pole RC filter except that the output from each previous pole is used as the input for the successive pole, where:

$$Y(n) = X(n_p) + W + (Y(n_p - 1) - X(n_p - 1))$$

for $p = 1$ to 8, where $Y(n)$ is the output from the current pole, $X(n_p)$ is the filter output of the previous pole for the current point, $Y(n_p–1)$ is the output from the previous pole for the previous point and:

$$W = e^{-dt/\tau} \qquad \text{where} \qquad \tau = 1/2\pi(f/f_N)$$

where f_N is the normalized cutoff frequency. With the coincident pole design the cutoff frequency rises as the order of the filter is increased. The normalized cutoff frequency, f_N, is given by:

$$f_N = 1/\sqrt{2^{1/n} - 1}$$

where *n* is the order (number of poles) of the filter. For an 8-pole filter this value is 3.32397. The specified cutoff frequency must be divided by the normalization factor in order to adjust the positions of the multiple poles. A consequence of this is that the maximum f_C/f_S ratio must be limited to the Nyquist frequency (f_C/f_S = 0.5) divided by the normalized cutoff frequency, or 0.5/3.32397 = 0.15.

Expected vs. Observed Rise Times

100 mV step pulse, f_S = 10 kHz

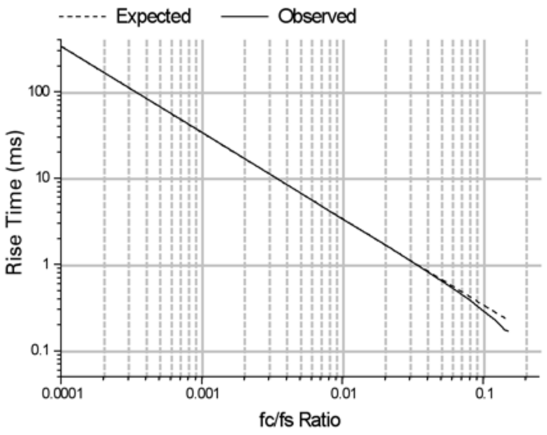


Figure 8.16: RC lowpass filter (8 pole) expected vs. observed rise time.

Normalized Frequency Response

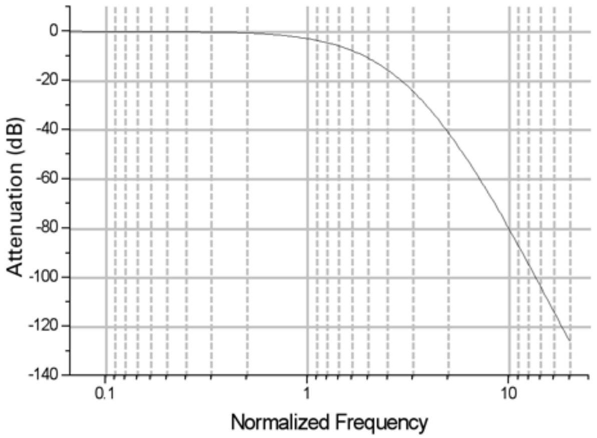


Figure 8.17: RC lowpass filter (8 pole) normalized frequency response.

RC HIGHPASS FILTER (SINGLE POLE) SPECIFICATIONS

→ Attenuation: 20 dB at $f = 0.1\ f_c$

The single-pole RC highpass filter operates by subtracting the lowpass response from the data. This is valid in this case because of the constant group delay characteristics of the single pole RC filter. The recurrence formula for this filter is, therefore:

$$Y(n) = X(n) - [X(n) + W + (Y(n-1) - X(n-1))]$$

where $Y(n)$ is the current output, $X(n)$ is the current data point, $Y(n–1)$ is the output for the previous point and:

$$W = e^{-dt/\tau} \quad \text{where} \quad \tau = 1/2\pi f$$

where dt is the sampling interval and f is the –3 dB cutoff frequency.

Normalized Frequency Response

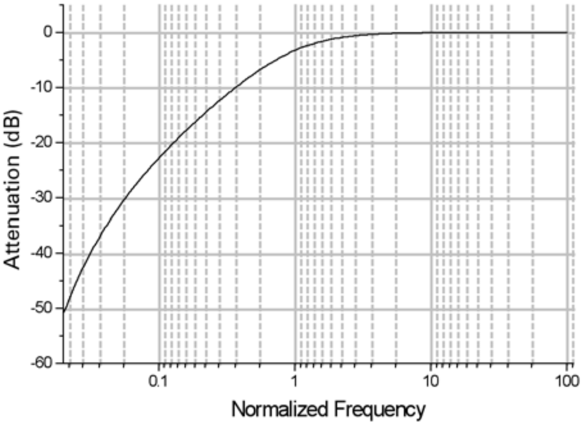


Figure 8.18: RC highpass filter (single pole) normalized frequency response.

BESSEL HIGHPASS FILTER (8-POLE ANALOG) SPECIFICATIONS

→ Attenuation: 114 dB at $f = 0.1\ f_c$

The highpass Bessel filter has a sharper roll-off than the highpass RC filter. However, the Bessel filter deviates from ideal behavior in that it introduces ringing in the response to a step function, as shown below.

Highpass Bessel Filter Step Response

100 mV step pulse (dotted line), f_S = 10 kHz, f_C = 100 Hz

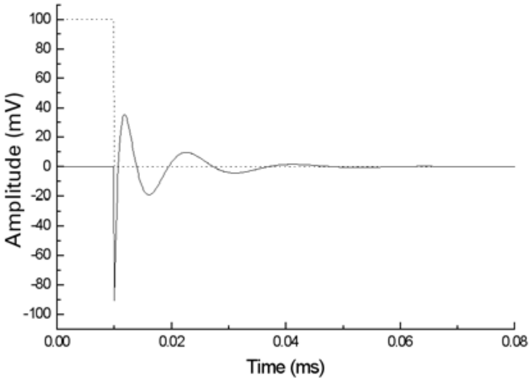


Figure 8.19: Highpass Bessel filter step response.

Normalized Frequency Response

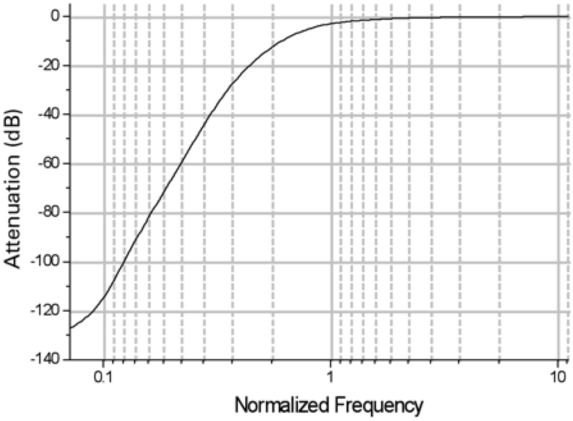


Figure 8.20: Highpass Bessel normalized frequency response.

THE ELECTRICAL INTERFERENCE FILTER

The electrical interference (EI) Filter removes power line interference from an acquired data signal. This filter identifies and removes complex power line waveforms composed of multiple harmonics. The interference detection method is adaptive; that is, the filter always matches the actual line signal even if its properties (frequency, phase and shape) change during measurement.

The core component of the EI filter is the sine detector that discriminates sinusoids from other additive signal components. The sine detector generates a reference sine wave and adjusts its phase until it locks to a specific line-interference harmonic. The EI filter uses an array of sine detectors, one for each harmonic of the interference waveform.

The sine detector basically operates as a digital Phase Locked Loop (PLL) tuned to line frequency (50/60 Hz) or its multiple. The correlator (phase detector) detects phase difference between the reference and actual line harmonic. This phase difference is used as a feedback signal to adjust the reference phase until a perfect match is achieved.

Reference signals, each locked to a specific interference harmonic, are subtracted from the original signal, thus canceling out the complete interference waveform.

Assumptions

1 The line interference and the data signal are statistically independent, i.e. uncorrelated.

2 The line signal $x(n)$ is stationary across at least M of its periods. We can assume that the line signal does not significantly change when observed at any M of its periods.

3 The measured signal $s(n)$ is the sum of data signal $y(n)$ with scaled and phase shifted (delayed) version of the line signal: $s(n) = y(n) + Ax(n–d)$

Problem Statement

We want to identify line signal $x(n)$ incorporated inside $s(n)$. We assume that line signal is composed of a certain number of sinusoids with harmonic frequencies. According to assumption 3, we need to determine A and d for each harmonic.

Basic Theory

In order to detect the line signal we use the fact that data signal $y(n)$ and line signal $x(n)$ are uncorrelated. Practically, we say that cross correlation R_{xy} is equal to zero:

$$R_{xy}(d) = \frac{1}{M\Delta}\sum_{i=0}^{M\Delta} y(i)x(i-d) = 0$$

for each d, where Δ is the number of samples per period of the line signal. In order to keep the argument as simple as possible, we will use the term “correlation” R_{xy} meaning in most places actually “covariance”, implicitly assuming all signals to have zero DC component.

The correlation of the line signal $x(n)$ and the measured signal $s(n)$ should equal to:

$$R_{xs}(d) = R_{xy}(d) + R_{xx}(d)$$

where R_{xx} is the auto-correlation of the line signal. Since R_{xy} is equal to zero, we conclude that if we correlate the line signal and measured signal, we will obtain the auto-correlation of the line signal. The auto-correlation function $R_{xx}(d)$ is even and its maximum is at d = 0. Furthermore, since $x(n)$ is periodic, R_{xx} is also periodic with the same period.

We use the above argument to determine both the delay *d* and the scale *A* of the line signal inside the measured signal (see assumption 3). The above discussion holds for our specific situation if we assume that *x*(*n*) is a sinusoidal reference signal.

EI Filter Block Diagram

The adaptive line interference filter block diagram for the fundamental harmonic is shown in Figure 8.21. The same procedure is repeated for each harmonic by multiplying the frequency of the reference generator.

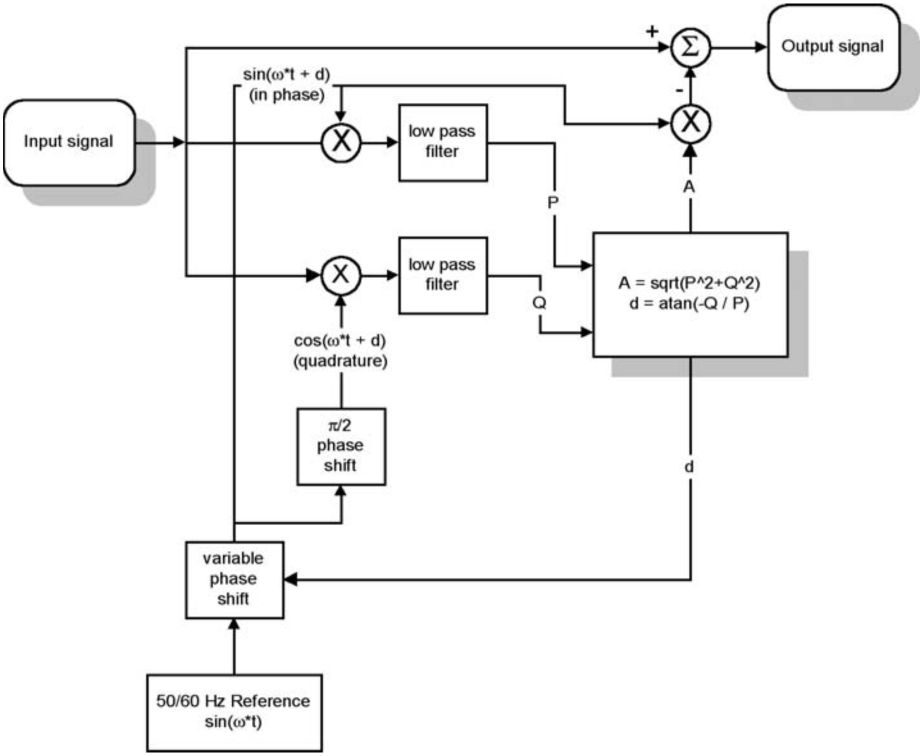


Figure 8.21: Adaptive line interference filter block diagram.

Implementation

The low pass filter is implemented as the simple average over *M* line periods. The average of the product of two signals, as in correlation definition, is equal to applying a low pass filter with box impulse response, or *sin*(*x*)/*x* transfer function.

The critical parameter for correlator performance is the averaging period *M*. The higher the value of *M*, the lower the bandwidth of the equivalent low pass filter. By keeping the bandwidth as low as possible, we reduce the correlation estimate error. Since we ultimately need to estimate the auto-correlation of the periodic signal *x*(*n*), the averaging period should be equal to a whole number of line signal periods *M*.

The EI filter generates a reference sine wave for each detected harmonic up to the maximum specified harmonic search number and subtracts reference waveforms from the signal, thereby canceling line interference. Ideally, each reference sinusoid exactly matches the corresponding interference harmonic both in phase and amplitude, resulting in the perfect cancellation of line interference.

A practical EI filter never exactly detects phase and amplitude of interference harmonics. After subtraction of the generated reference sinusoids any discrepancy in amplitude and phase will result in artifact sinusoids, that is, components that were not present in the original signal. After filtering, artifactual components actually replace the original line harmonics. When harmonic detection is good, the total power of artifact components after filtering is much lower than line interference power in the original signal.

Harmonic detection errors come from noise and data signal components at line harmonic frequencies (multiples of 50/60 Hz). Generally, noise errors are less significant and can be successfully reduced by increasing the number of cycles to average. A more significant source of EI-filter errors are original data signal components that fall at or close to 50/60 Hz multiples (data signal leak).

Weak Harmonics

Weak line harmonics in the presence of noise cannot be accurately detected. In extreme cases EI-filtering artifact power for the single weak harmonic can be larger than the actual harmonic power. In such cases it might be better to reduce the harmonic search number in order to exclude weak harmonics. Also, increasing the number of cycles to average will always improve harmonic detection (if the noise is the main error source). However, excessively large number of cycles to average will negatively affect execution speed, tracking performance and startup transient compensation.

Data Signal Components

Periodic

Any periodic components in the data signal with frequencies at or very close to 50/60 Hz multiples will leak through the EI filter harmonic detectors and will produce false line interference harmonics.

Figure 8.22 shows the result of filtering a pure (no noise and no line interference) square wave at 10 Hz with the harmonic search number set to 3 and reference frequency set to auto.

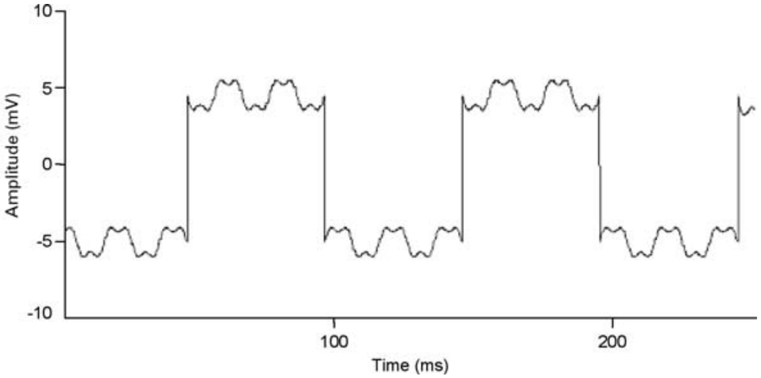


Figure 8.22: Results of filtering a square wave.

Since a 10 Hz square wave has strong components at multiples of 10 Hz, the EI filter locked on to the 5th (50 Hz), 10th (100 Hz) and 15th (150 Hz) harmonics, generating prominent artifacts.

Aperiodic

Strong and sharp pulses in the data signal may produce artifacts in the EI filtering process. Sharp pulses (spikes) have significant components at all frequencies including 50/60 Hz multiples. If the spike amplitude is two (or more) orders of magnitude larger than actual line interference amplitude, the EI filter will produce false line harmonics after the spike in the region whose size is equal to the number of cycles to average.

In the example in Figure 8.23, the EI filter was used at 10 harmonics with 50 cycles to average. Spike amplitude (not shown in full) was more than 200 mV. Notice the false line harmonics in the lower graph.

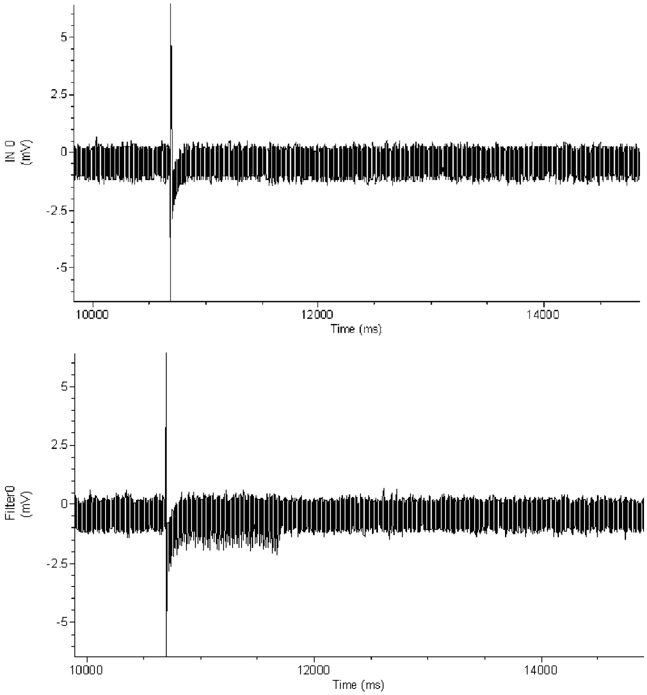


Figure 8.23: EI filtering showing introduction of false line harmonics.

Start-up Transients

Start-up transients are spurious, rapidly changing false harmonics at the very beginning of the filtered signal. When processing samples at the beginning of the signal file EI filter does not have enough information to accurately detect line harmonics. With every new processed sample the detection becomes better and better and the start-up transient becomes smaller and smaller. The filter reaches its steady state after a time equal to number of cycles to average.

The EI filter automatically compensates for start-up transients by turning off reference subtraction until it reaches its steady state. When it reaches steady state after the specified number of cycles to average the EI filter assumes that line interference parameters are accurately detected and filters the signal backwards using current reference parameters.

Potential Problems

The filter is too slow

When dealing with large datasets and high sampling rates the filter might be slow. If filtering is unacceptably slow try the following:

- Check if it is necessary to remove all harmonics specified in the harmonics field. Try removing only the first harmonic, and if the result is not satisfactory, increase the harmonic number and try again. The removal of the first three harmonics will often sufficiently reduce the interference.

- Decrease the value in the Cycles to average field. Often, smaller averaging lengths (time constants) do not significantly affect the output signal quality.

Interference is not fully removed

If the line interference is not fully removed try the following:

- If residual interference contains high frequencies then it is might be necessary to increase the value of the upper harmonic to be removed.
- If the fundamental line harmonic is still visible in the output signal then the number of cycles to average should be increased.

Cutoff Frequency Limitations

All digital filters have inherent limitations, and in some cases deviate significantly from their analog counterparts. The filters in Clampfit have been restricted to an f_C/f_S ratio where the filter response is reasonably close to the theoretically expected response.

- The theoretical frequency range of digital filters is between 0 and the Nyquist frequency, which is one-half of the sampling frequency. This applies to all filters when filtering sampled data. However, the usable range of most software filters is considerably narrower than this theoretical range. The usable range depends on the nature of the filter (FIR or IIR) and the filter algorithm.
- The overshoot during a step response is a characteristic feature of Bessel, Butterworth and Chebyshev lowpass filters. For analog filters, the magnitude of the overshoot is constant over the full operating range. For digital IIR filters, however, the overshoot becomes increasingly larger as the ratio of f_C/f_S increases.
- The operating range of the Gaussian FIR filter is limited at the low end by a practical, rather than theoretical, limitation. Low ratios f_C/f_S result in the generation of a large number of filter coefficients. This creates two problems. The first is that smaller datasets cannot be accurately filtered because the filter length might be greater than the number of data points. The second is that the large number of coefficients is computationally inefficient. The number of Gaussian coefficients is inversely proportional to the f_C/f_S ratio where a lower cutoff frequency requires a greater number of coefficients for the filter realization.

The following table lists the numerical limitations for each filter type. The lower and upper cutoff frequencies are expressed as a factor times the sampling frequency (f_S). These limits are internally set by the filter algorithm and cannot be altered.

Table 8.1: Numerical limitations for each filter type.

Filter Type	Lower Cutoff Limit	Upper Cutoff Limit
Bessel (8-pole IIR)	10^{-4} x f_S	0.14 x f_S
Boxcar (FIR)	See note 1.	n/a
Butterworth (8-pole IIR)	10^{-4} x f_S	0.2 x f_S
Chebyshev (8-pole IIR)	10^{-4} x f_S	0.2 x f_S
Electrical Interference	See note 2.	n/a
Gaussian (FIR)	10^{-4} x f_S. See note 3.	0.5 x f_S
Notch (2-pole IIR)	10^{-3} x f_S	0.3 x f_S
RC (single-pole IIR)	10^{-4} x f_S	0.5 x f_S
RC (8-pole IIR)	10^{-4} x f_S	0.15 x f_S. See note 4.

Note 1

The boxcar filter requires that the number of smoothing points be specified. This must be an odd number in order for the filter to be symmetrical. The minimum number of smoothing points is 3. The maximum number of smoothing points is 99. However, the maximum number of smoothing points is also limited by the number of data points, *n*, such that the filter width is at least *n*/2. So if there are 50 data points the maximum number of smoothing points is 50/2 = 25. If this formula generates an even number then the maximum number of smoothing points will be one less. For example, if there are 52 data points, then the maximum number of smoothing points will be 52/2 – 1 = 25.

Note 2

The electrical interference filter does not have a lower or upper cutoff frequency limit as it is preset to remove either 50 Hz or 60 Hz interference and the associated harmonics (see *"The Electrical Interference Filter"* on page 148). However, there is a data point minimum, as this filter requires a specific number of points to reach steady state. This minimum is given by:

$$\textit{minimum points} = \textit{samples per period} \times \textit{cycles to average}$$

where the samples per period is the sampling frequency divided by the reference frequency (50 Hz or 60 Hz) and cycles to average is the number of cycles of the reference frequency which are averaged in the response. For example, for a sampling rate of 1 kHz, a reference frequency of 60 Hz and 20 cycles to average the minimum number of data points required is 1000/60 x 20 = 334 data points.

8. Digital Filters

Note 3
The Gaussian filter width (see *"Finite vs. Infinite Impulse Response Filters"* on page 131) depends on the f_c/f_S ratio; the lower this ratio the greater the number of Gaussian coefficients (see *"Gaussian Lowpass Filter Specifications"* on page 141). In view of this, two criteria are used to limit the lower cutoff frequency.

The first is that there must be enough data points to accommodate at least two Gaussian filter widths. That is, the minimum corner frequency will correspond to a filter width that is less than or equal to one-half the number of available data points.

The second is that the maximum number of Gaussian coefficients is limited to approximately 3500. This limit, which corresponds to an f_c/f_S ratio of about 3 x10^{-4}, is not exact because the automatically computed minimum corner ratio is generally rounded up. Therefore, the minimum corner ratio might correspond to a number of coefficients that is somewhat more or less than the 3500 limit.

Note 4
The 8-pole RC filter is a "multiple coincident pole" design where the –3 dB cutoff frequency rises with each pole by an amount given by:

$$f_N = 1/\sqrt{2^{1/n} - 1}$$

where f_N is the normalized cutoff frequency and n is the number of poles. Therefore, for an 8-coincident-pole filter the normalized cutoff frequency is actually 3.32397 times the specified cutoff frequency (see *"RC Lowpass Filter (8 Pole) Specifications"* on page 145). Consequently, the maximum f_c/f_S ratio must be limited to the Nyquist frequency (f_c/f_S = 0.5) divided by the normalized cutoff frequency, or 0.5/3.32397 = 0.15.

9. Clampfit Analysis

Digital spectral analysis involves the decomposition of a signal into its frequency components. The purpose of such analysis is to reveal information that is not apparent in the time-domain representation of the signal. To this end the Fast Fourier Transform (FFT) is generally used because of its speed. The Fourier transform is based on the concept, first advanced by Jean Baptiste Joseph, Baron de Fourier, that nonperiodic signals can be considered as the integral of sinusoids that are not harmonically related.

The FFT samples in the frequency domain, just as a digital signal is sampled in the time domain. A signal processed by the FFT yields a set of spectral coefficients that can be regarded as samples of the underlying continuous spectral function. Although long transforms might look like a continuous spectrum, in fact they consist of discrete coefficients that define the “line spectrum” of the signal. The line spectrum indicates the “amount” of the various frequencies that are contained in the signal.

THE FOURIER SERIES

For a periodic digital signal x, the coefficients of the spectral distribution can be represented by the Fourier Series, which is defined by:

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad (1)$$

where, for samples $x = x_0$ to x_{N-1}, a_i is the k^{th} spectral component, or harmonic, j is a complex number and N is the number of sample values in each period of the signal. The real and imaginary parts of the coefficients generated by this function can be expressed separately by writing the exponential as:

$$cos(2\pi kn/N) - j\,sin(2\pi kn/N)$$

If $x[n]$ is a real function of n, the coefficient values except for a_0 are symmetrical. The real parts of coefficient a_0 are unique, but the real parts of coefficients a_1 and a_{N-1} are identical, as will the real parts of a_2 and a_{N-2}, and so forth. Likewise the imaginary parts also follow this pattern except that the sign changes so that $a_1 = -a_{N-1}$, and so forth. The imaginary part of coefficient a_0 (the zero-frequency coefficient) is always zero.

This symmetry extends to coefficients outside the range 0 to N–1 in both the positive and negative direction. For example, coefficients a_N to a_{2N-1} are identical to coefficients a_0 to a_{N-1}. Thus periodic digital signals have spectra that repeat indefinitely along the frequency axis. This is also true for nonperiodic signals that are more likely to be encountered in the real world (see next section, *"The Fourier Transform"*).

A periodic digital signal with N samples per period can, therefore, be completely specified by in the frequency domain by a set of N harmonics. In fact, if $x(n)$ is real then only half this number is required because of the inherent symmetry in the coefficient values.

Fourier analysis is concerned with determining the amount of each frequency that is present. This appears ambiguous for a digital signal since a whole set of spectral representations is possible. However, this is not a problem because spectra that are produced by digital Fourier analysis repeat indefinitely along the frequency axis, so only the first of these repetitions is sufficient to define the frequency content of the underlying signal, so long as the Sampling Theorem is obeyed (see *"The Sampling Theorem in Clampfit"* on page 15).

If the "transform length" N contains an integral number of cycles then the natural periodicity of the signal is retained and each component of the spectrum occupies a definite harmonic frequency. This is because the natural periodicity of each component is preserved during spectral analysis. However, if $x[n]$ contains sinusoids that do not display an exact number of periods between 0 and N then the resultant spectrum displays discontinuities when repeated end on end. Because of these discontinuities the spectrum displays a spreading of energy, or "spectral leakage". In the real world, signals rarely if ever are so cooperative as to arrange themselves in integral numbers of cycles over a given transform length. Thus spectral leakage will, in general, always be a problem. Fortunately, there are procedures that can minimize such leakage (see *"Windowing"* on page 160).

THE FOURIER TRANSFORM

Most practical digital signals are not periodic. Even naturally occurring repetitive signals display some degree of variation in frequency or amplitude. Such signals are evaluated using the Fourier Transform, which is closely related to the Fourier Series defined by Equation 1.

Equation 1 applies strictly to a periodic signal with a period N. However, we can modify this equation to apply to nonperiodic signals. Starting with a periodic signal of length N, we select an arbitrary number of points in the center of the signal. We then "stretch" the signal by adding zeros to either side of the selected points. We can continue adding zeros until $N \rightarrow \infty$. At this point the neighboring repetitions have moved to $\pm\infty$ and we are left with a nonperiodic signal.

If the signal is stretched in this way then the coefficients, a_k, must become smaller because of the $1/N$ term in Equation 1. They must also come closer together in frequency because N also appears in the denominator of the exponential. Therefore, in the limit as $N \rightarrow \infty$, the various harmonics become spaced extremely closely and attain vanishingly small amplitudes. We can think of this in terms of a continuous, rather than a discrete, spectral distribution.

As $N \to \infty$ the product Na_k remains finite although each spectral coefficient, a_k, becomes vanishingly small. We can write Na_k as X. Also, the term $2\pi k/N$ can be thought of as a continuous frequency variable that can be written as Ω. Thus, Equation 1 becomes:

$$x = Na_k = \sum_{n=0}^{N-1} x[n]e^{-j\Omega n} \tag{2}$$

Since $x[n]$ is now nonperiodic the limits of summation should be changed. Also, since $x[n]$ exists for both positive and negative values of n we sum between $n = \pm\infty$. We also write X as $X(\Omega)$ to make it clear that X is a function of the frequency Ω. Therefore, Equation 2 becomes:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$$

which defines the Fourier Transform of the nonperiodic signal $x[n]$.

Just as for periodic signals, the spectrum of a nonperiodic digital signal is always repetitive.

THE FAST FOURIER TRANSFORM

Clampfit uses the Fast Fourier Transform (FFT) for spectral decomposition. Computing the Fourier Transform is a computationally intensive task that requires a great deal of redundant calculation. The Fast Fourier Transform is an algorithm that reduces these redundant calculations, in part by decomposing the Discrete Fourier Transform (DFT) into a number of successively shorter, interleaved DFTs.

For example, if we have signal $x[n]$ with N points where N is a power of 2 then we can separate $x[n]$ into two short subsequences, each of length $N/2$. The first subsequence contains the even-numbered points and the second contains odd-numbered points. Successive decomposition of an $N/2$-point subsequence can be broken down until each sequence is a simple 2-point DFT. This process is known as “decimation in time”. Computing the Fourier Transform at this point is still not trivial and requires factors to account for the displacement of the points in time.

For a more complete discussion of the FFT see Lynn and Fuerst 1994.

Clampfit uses a decimation in time FFT algorithm that requires N to be an integral power of 2.

THE POWER SPECTRUM

The magnitude of the spectral components, S, derived by the Fourier transform is given by the sum of the squares of the real and imaginary parts of the coefficients, where:

$$S_i = a_{i(real)}^{\;2} + a_{i(imag)}^{\;2}$$

Since this is the power averaged over N samples (where N is the transform (window) length) and since only $N/2$ of the components are unique, the power at each sampling point (P_i) is scaled such that, for a given sampling frequency, f:

$$P_i = S_i\frac{2N}{f}$$

The power spectrum is further scaled by the window factor, ϖ, where for a given window function, f_w, the scale factor for a window of length N is given by:

$$\varpi = \frac{\sum_{n=1}^{N} (f_w)^2}{N}$$

and the power, expressed in units^2, is given by:

$$P_i = S_i\frac{2N}{f\varpi}$$

Finally, the total root mean square (RMS) value is computed by taking the square root of the integral of the power spectrum, such that:

$$RMS = \sqrt{\sum_{n=1}^{N} P_i f/N}$$

where f/N is value of the frequency bin width in Hz.

LIMITATIONS

The limitations imposed on power spectral analysis are:

- The data must be sampled at evenly spaced intervals.
- The transform length must be an integral power of 2.
- The frequency spectrum will range from 0 to one-half the sampling frequency.

WINDOWING

If all frequency components in a signal have an integral number of periods in the transform length then each component occupies a definite harmonic frequency, corresponding to a single spectral coefficient. However, real-world signals are rarely so cooperative, containing a wide mixture of frequencies with few exact harmonics. Thus spectral leakage (see *"The Fourier Series"* on page 157) is almost always expected. In order to reduce spectral leakage it is common practice to "taper" the original signal before transformation to reduce edge discontinuities. This is done by multiplying the signal with a "windowing function".

The ideal window has a narrow main lobe to prevent local spectral leakage and very low side lobe levels to prevent more distant spreading. As it turns out, these two requirements are in conflict. Consequently all windowing types allow some spectral leakage.

The simplest window is the rectangular (or "do-nothing") window that does not alter the signal. This window has the narrowest possible main lobe but very large side lobes, which permit a substantial amount of spectral leakage with nonharmonics (there is no leakage of exact harmonic components). All other windowing functions taper the data to a greater or lesser degree.

Clampfit offers several window functions. The choice of a windowing type depends on the nature of the signal and the type of information that is to be extracted. To assist in this decision the window function is displayed in the dialog box along with the time domain and frequency domain responses for each window type.

SEGMENT OVERLAPPING

Data can be divided into segments of equal length either for sequential processing to reveal a trend in the power spectra with time or to average the power spectra for all segments in order to reduce the variance. If the spectra are to be averaged then segmenting can either be overlapped or the segments can be generated without overlapping. Both options are available in Clampfit.

Note that the "segments" are equal to the window length, which in turn is equal to the FFT transform length.

If the segments are not overlapped then the reduction in the spectral variance is about $M/2$ where M is the number of segments. If the segments are overlapped by 50% then the reduction in spectral variance is about $9M/11$, which is a significant improvement over the non-overlapped case (Press *et al.* 1992). For example, if 10 segments are averaged without overlapping the spectral variance is reduced by a factor of about 5. However, with 50% overlapping the variance is reduced by a factor of about 8.2.

TRANSFORM LENGTH VS. DISPLAY RESOLUTION

The sampling theorem states that the highest frequency that can be contained in a digital record is equal to one-half the sampling frequency (see *"The Sampling Theorem in Clampfit"* on page 15). Therefore, the highest frequency component that can be resolved by Fourier analysis is also only one-half the sampling frequency. Moreover, the symmetry inherent in the spectral coefficients means that only one-half of the coefficients are required to completely define the frequency spectrum of the digital signal. These issues are reflected in the display of the power spectrum in Clampfit where the scale ranges from 0 to one-half the sampling frequency. A two-sided display would simply show a symmetrical (mirror image) distribution of spectral lines.

The transform (window) length relative to the sampling frequency determines the resolution of the power spectrum. In view of the restriction imposed by the frequency resolution, the frequency scale (X axis) for a one-sided display ranges from 0 to $f_S/2$, where f_S is the sampling frequency. The resolution is dependent on the transform length L. For example, if f_S is 10 kHz and L is 512 then the frequency scale, ranging from 0 to 5000 Hz, is divided into 256 bins (only one-half the transform length is used because of the transform symmetry), each having a width of 19.53 Hz (5000 Hz/256). If, on the other hand, L is 56 then the frequency scale is divided into 28 bins, each with a width of 178.57 Hz.

The bin width, W, in the spectral display is, therefore, given by:

$$W = f_S / L$$

10. pCLAMP Analyses

MEMBRANE TEST

The Membrane Test in Clampex (Tools > Membrane Test) generates a definable voltage pulse and reads off a range of measurements from the current response:

- Membrane capacitance (C_m)
- Membrane resistance (R_m)
- Access resistance (R_a)
- Time constant (Tau)
- Holding current (Hold)

Membrane Test applies a continuous square wave voltage command and the current response is measured. Voltage-clamp mode is assumed.

Command Pulse

Each pulse period uses 500 samples, corresponding to 250 samples per step with a 50% duty cycle.

The command pulses are measured relative to the holding level. The pulse height is expressed as peak-to-peak (p-p). Both edges of the pulse (i.e. both capacitive transients) are used for calculations.

A fast logarithmic exponential fit is performed on each transient or each averaged transient using a look-up table for the log transforms. The fit is performed between the 10% to 80% ordinates, or other ordinates, according to the settings in the Proportion of Peak to Fit section in the Options dialog. The fit is displayed on the raw or averaged data as a superimposed red line.

Calculations

The transient portion of the current response is fit by a single exponential. From this exponential and the area under the curve the following parameters can be determined:

- The steady-state current (I_{ss} or HOLD)
- The time constant of the transient (τ or Tau) and the peak response
- The electrode access resistance (R_a)
- The membrane resistance (R_m)

→ The membrane capacitance (C_m).

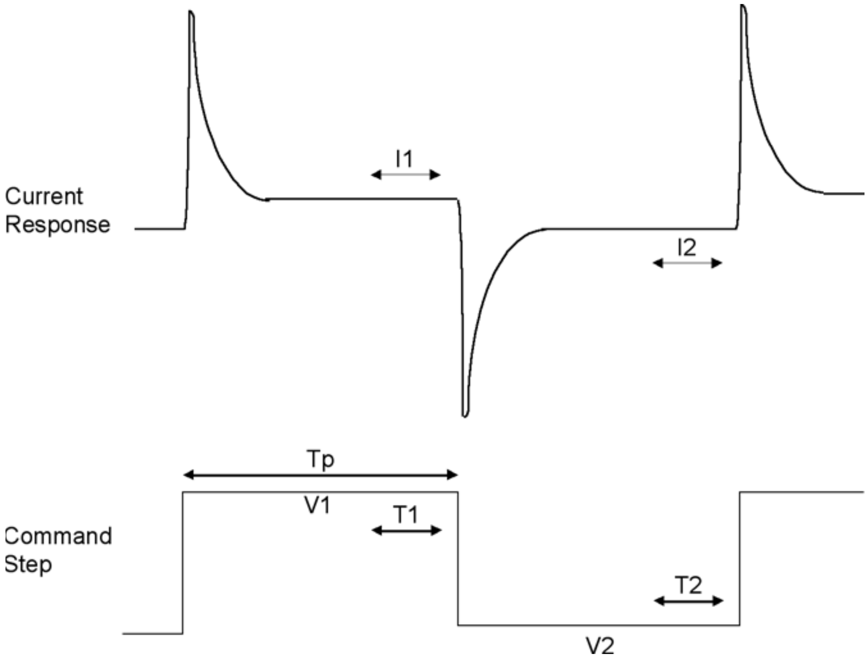


Figure 10.1: Regions of current response used in Membrane Test calculations.

The average current (I_1) is measured during period T_1, which is 20% of the duration of T_p. The average current (I_2) of the second pulse in the pair is measured during period T_2, and is the baseline for the first pulse. The average current (I_1) of the first pulse in the pair is the baseline for the second pulse.

When calculating the charge under the transient, the settling time of the membrane voltage step is corrected by adding $\Delta I \times \tau$ (Q_2; see below) to the integral, where $\Delta I = I_1 - I_2$. The steady-state current (I_{ss}) is calculated as the average of I_1 and I_2 [= (I_1 + I_2)/2] (see Figure 10.2).

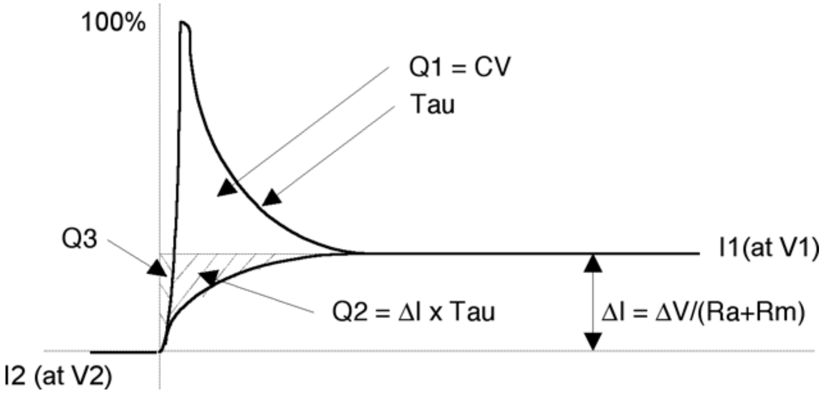


Figure 10.2: Derivation of Membrane Test results.

The total charge (Q_t) under the transient is the sum of the charge (Q_1) in the transient above the steady-state response plus the correction factor (Q_2). The small error charge (Q_3) introduced in the calculation of Q_2 is ignored. A logarithmic fit is used to find the time constant (τ) of the decay. Q_1 is found by integrating the charge above I_1. When integrating to find Q_1, I_1 is subtracted from I_t before integrating.

$$Q_2 = \Delta I \times \tau$$

$$Q_t = Q_1 + Q_2$$

$$C_m \text{ is derived from } Q_t = C_m \times \Delta V_m \qquad (1)$$

where ΔV_m is the amplitude of the voltage change across the capacitor, i.e. the change in the membrane potential. In the steady-state, the relation between ΔV_m and ΔV is:

$$\Delta V_m = \Delta V \times R_m / R_t = \Delta V \times R_m / (R_a + R_m) \qquad (2)$$

where R_t, R_m and R_a are the total, membrane and access resistances, respectively. Substituting for ΔV_m from (2) into (1) derives:

$$C_m = Q_t \times R_t / \Delta V \times R_m \qquad (3)$$

Solving for R_a

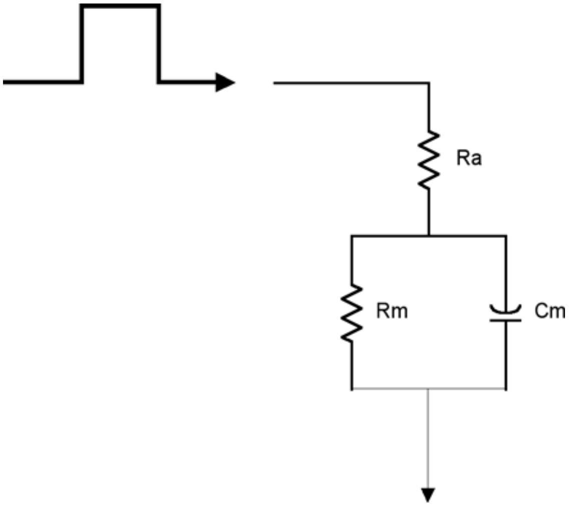


Figure 10.3: Idealized cell circuit.

The circuit elements R_a, R_m and C_m are readily derived as follows. Substituting (3) for C_m in the definition of the time constant = $R_a \times C_m$:

$$\tau = Q_t \times R_a / \Delta V$$

hence:

$$R_a = \tau \times \Delta V / Q_t \tag{4}$$

which provides access resistance directly as a function of measured variables. The total resistance is calculated from the steady-state response:

$$R_t = \Delta V / \Delta I = R_a + R_m \tag{5}$$

from which one obtains:

$$R_m = R_t - R_a \tag{6}$$

C_m is then obtained from (3):

$$C_m = Q_t \times R_t / \Delta V \times R_m \tag{3}$$

When calculating the response for the second transient (the downward pulse), the same current measurements as the upward pulse (I_2 <–> I_1) are used and the second pulse is inverted in order to use the same calculation code.

[We thank Dr. Fernando Garcia-Diaz of the Boston University School of Medicine for his suggestions for these calculations.]

TEMPLATE MATCHING

In event detection template matching in Clampfit, (Event Detection > Template Search) the template is slid along the data trace one point at a time and scaled and offset to optimally fit the data at each point. Optimization of fit is found by minimizing the sum of squared errors (SSE) between the fitted template and the data. A detection criterion value is calculated at each position, by dividing the template scaling factor by a measure of the goodness of fit (derived from the SSE). You are able to set a threshold value for this criterion in the Template Search dialog, “Template match threshold”. All sections of the trace with a detection criterion greater than or equal to this value are automatically offered as candidate events.

When the template is aligned with an event the detection criterion is closely related to the signal-to-noise ratio for the event. Since background noise rarely exceeds four times the standard deviation of the noise, an (absolute) detection criterion value greater than four is unlikely unless there is an underlying event. For most data, then, four provides a close to optimum setting for the template match threshold value. Settings less than this run the risk of making false positives, while settings greater than this may miss genuine events.

For further information see Clements and Bekkers, 1997.

SINGLE-CHANNEL EVENT AMPLITUDES

Filtering data reduces the amplitude of rapidly occurring events (rapid with respect to the filter). In particular, short single-channel events have an attenuated amplitude. In addition, the points at the ends of long single-channel events have a smaller amplitude than in the absence of the filter.

When Clampfit determines the amplitude of a single-channel event, it takes the average of all data points not affected by the filter. Data points within two time constants of the ends of each event are not included in the average. The time constant used takes into account the combination of digital filtering applied in Clampfit prior to the single-channel search being run, and the acquisition filters, as follows.

$$n_f = \frac{1}{2\pi\Delta t}\sqrt{\frac{1}{f_{Inst}^2} + \frac{1}{f_{CyberAmp}^2} + \frac{1}{f_{Postacq}^2}}$$

where n_f is the settling time in sample points, t is the sampling period, f_{Inst}, and $f_{CyberAmp}$ are the –3 dB cutoff frequencies of the Instrument (or external) and CyberAmp filters used during acquisition, and $f_{Postacq}$ is the –3 dB cutoff frequency of Clampfit’s digital filter.

If a single-channel event is so short that all of its points are affected by the filter, no averaging is performed: the midpoint of the channel event is taken to be its amplitude, and “B” (for “brief”) is written into the Results sheet State column for the event.

Note
Failing to specify correctly the instrument filter used (if any) when acquiring data invalidates the calculation of single-channel event amplitudes by Clampfit. The utility program header.exe can be used to correct data files that contain an incorrectly specified acquisition filter frequency.

LEVEL UPDATING IN SINGLE-CHANNEL SEARCHES

To accommodate baseline drift in single-channel detection, the single-channel search dialog allows you to specify automatic level updating, applied as a search proceeds. You can opt to update the baseline only, without altering the relative amplitudes of the other levels, or update all levels independently.

When you select this option you must set the amount that new events contribute to their level's running amplitude. This weighting is entered as a percentage. With the default level contribution of 10%, for example, a new event contributes 10% of its amplitude, and the previous value, 90%, to the new level. For example, with a previous value of 10.0 pA, a new event with an amplitude 10.5 pA, and 10% contribution, the new amplitude for the given level is:

$$(0.9 \, x \, 10.0 \, pA) + (0.1 \, x \, 10.5 \, pA) = 10.05 \, pA$$

The algorithm also includes a means to weight short events less, so that rapid events influence the level less than the level contribution you have stipulated. This is in part because short events, affected by signal filtering, return amplitudes less than actually occurred (amplitudes of longer events are measured from the central portion of the event, avoiding the filter-affected transition periods at the start and end of the event). Because the reduced weighting of short events is designed to mitigate filtering effects, the amount of filtering a signal has undergone is used to determine the length of "brief" events, i.e. the point at which reduced weighting begins to apply. An event is classified short when it is shorter than 50 sample points multiplied by two time constants. As for single-channel event amplitudes (the previous section), two time constants is the time equivalent of n_f, for the number of samples, in the following equation:

$$n_f = \frac{1}{2\pi\Delta t}\sqrt{\frac{1}{f^2_{Inst}} + \frac{1}{f^2_{CyberAmp}} + \frac{1}{f^2_{Postacq}}}$$

Here, t is the sampling interval, and f_{Inst}, $f_{CyberAmp}$, and $f_{Postacq}$ the –3 dB cutoff frequencies of the instrument (i.e. amplifier), CyberAmp and Clampfit digital filters respectively.

As an example of the application of this, in a file recorded at 400 µs sampling interval (2500 Hz), with 500 Hz lowpass filter setting, any event with less than 40 samples (16 ms) qualifies as short. The contribution such events make to the level update is directly proportional to the extent the event is shorter than this, e.g. with a level contribution setting of 10%, an event lasting 8 ms (i.e. half a "short" event for these sampling and filtering rates) contributes 5% to the new level. On the other hand, all events over 16 ms affect the level update equally, at 10%.

KOLMOGOROV-SMIRNOV TEST

The two-sample Kolmogorov-Smirnov (K-S) test is used to compare two independent "samples" (or "datasets") to test the null hypothesis that the distributions from which the samples are drawn are identical. For example, the K-S test can be used to compare two sets of miniature synaptic event amplitudes to see if they are statistically different. Student's t-test (which assumes data are normally distributed) and the nonparametric Mann-Whitney test (which does not) are sensitive only to differences in means or medians. The K-S test, however, is sensitive to a wider range of differences in the datasets (such as changes in the shape of the distributions) as it makes no assumptions about the distribution of the data. This wider sensitivity comes at the cost of power-the K-S test is less likely to detect small differences in the mean that the Student's t-test or Mann-Whitney test might otherwise have detected.

The K-S test should be used on data that are not normally distributed. If the data conform to a normal distribution, Student's t-test is more sensitive.

The test statistic ("K-S statistic", or "*D*") is the largest vertical (Y axis) difference between cumulative histograms derived from the datasets. The *p* value represents the probability that the two samples could have come from the same distribution by chance. In the K-S test, as in other tests, a $p < 0.05$ is typically accepted, by convention, as "statistically significant"—the samples are significantly different.

As part of the analytical process, this test measures the maximum value of the absolute difference between two cumulative distributions. This value is referred to as the Kolmogorov-Smirnov *D* statistic (or the "K-S statistic"). Thus, for comparing two different cumulative distribution functions, $S_{N1}(x)$ and $S_{N2}(x)$, the K-S statistic is:

$$D = (-\infty < x < +\infty)\left|S_{N_1}(x) - S_{N_2}(x)\right|^{max}$$

The calculation of the significance of the null hypothesis that the samples drawn from two distributions belong to the same population uses a function that can be written as the following sum:

$$Q_{KS}(\lambda) = 2\sum_{j=1}^{\infty}(-)^{j-1}e^{-2j^2\lambda^2} \qquad (1)$$

which is a monotonic function with the limiting values:

$$Q_{KS}(0) = 1 \qquad Q_{KS}(\infty) = 0$$

In terms of Equation 1, the significance level p of an observed value of D, as an evaluation of the null hypothesis that the distributions are the same, is approximated by the formula:

$$P(D > observed) = Q_{KS}\left(\left\lfloor \sqrt{N_e} + 0.12 + \frac{0.11}{\sqrt{N_e}} \right\rfloor D\right) \quad \text{(Stephens, 1970)}$$

where N_e, the effective number of data points, is given by:

$$N_e = \frac{N_1 N_2}{N_1 + N_2}$$

where N_1 is the number of points in the first distribution and N_2 is the number of points in the second distribution (Press *et al.* 1992). The closer the value of p is to 1, the more probable it is that the two distributions come from the same population.

Clampfit supports both “one-dimensional” and “two-dimensional” K-S tests. In the case of the one-dimensional test, the values of the data samples that are to be compared are not altered prior to the application of the test. One simply selects two different sets of data and applies the K-S statistic. In the case of the two-dimensional test, the data from the two populations are binned into a cumulative histograms (either fixed or variable width), and the binned distributions are subsequently evaluated using both the x values (bin widths) and y values (bin counts).

NORMALIZATION FUNCTIONS

Normalization functions are located in five locations in Clampfit. These are listed below.

Normalize Traces

Analyze > Normalize Traces is used to normalize trace data in the Analysis window. It offers two normalization options:

Use All Points

The traces are adjusted such that all points span the range 0 to 1. The normalization function for each point y in the trace is:

$$f(y) = (y - y_{min})/(y_{max} - y_{min})$$

where y_{min} is the smallest value in the trace and y_{max} is the largest value in the trace.

Use Specified Regions

The traces are adjusted such that the mean of the “Baseline region” is at 0 and the traces peak at +1.0 or –1.0 in the “Peak region”. The normalization function for each point y in the trace is:

$$f(y) = (y - b_y)/p_{|y|}$$

where b_y is the arithmetic mean of the points in the specified baseline region and $p_{|y|}$ is the largest absolute excursion between the mean baseline value and the specified peak region, where, for points n in the peak region:

$$p_{|y|} = max|y_n - b_y|$$

If the largest excursion from b_y in the peak region is to a value more negative than b_y, the normalized trace will peak at –1.0 in the peak region. If the largest excursion from b_y in the peak region is to a value more positive than b_y, the normalized trace will peak at +1.0 in the peak region. Note that if the response in the peak region contains values both above and below b_y, only the polarity that contains the largest absolute excursion from b_y will span a range of 1.0.

Normalize

Analyze > Normalize is used for normalizing data in Graph windows. Either the area under the plot is normalized, or the Y axis range.

Normalize Area

Plots are adjusted such that the area under the curves is equal to 1. The normalization function for each point y in the plot is:

$$f(y) = y/|y_{total}|$$

where $| y_{total} |$ is the sum of the absolute y values in the plot. This feature is primarily intended for normalizing the area under histograms.

Normalize Range

The range option adjusts the points so that all points span the range 0 to 1. The normalization function for each point y in the trace is:

$$f(y) = (y - y_{min})/(y_{max} - y_{min})$$

where y_{min} is the smallest value in the trace and y_{max} is the largest value in the trace.

The Norm() Function in Trace Arithmetic

Data file traces in the Analysis window can be normalized using Analyze > Arithmetic. The norm() function adjusts the trace such that all points span the range 0 to 1. The normalization function for each point y in the trace is:

$$f(y) = (y - y_{min})/(y_{max} - y_{min})$$

where y_{min} is the smallest value in the trace and y_{max} is the largest value in the trace.

The Norm() Function in Column Arithmetic

Data in Results window columns can be normalized using Analyze > Column Arithmetic. The norm() function adjusts the column data such that all points span the range 0 to 1. The normalization function for each point y in the column is

$$f(y) = (y - y_{min})/(y_{max} - y_{min})$$

where y_{min} is the smallest value in the trace and y_{max} is the largest value in the trace.

Normalize Histogram Area

When histograms are created from Analysis, Graph and Results window data using Analyze > Histogram, you have the option of creating a histogram with the area under the plot normalized.

The normalization function for each point y in the plot is:

$$f(y) = y/|y_{total}|$$

where $| y_{total} |$ is the sum of the absolute y values in the plot.

VARIANCE-MEAN (V-M) ANALYSIS

Variance-mean analysis (V-M) is a new analytical method, developed by Clements and Silver (1999), to quantify parameters of synaptic function.

The starting point for V-M analysis is the premise that transmission at a synapse can be described by three parameters. The first is the average amplitude of the postsynaptic response to a packet of transmitter (Q), the second is the number of independent presynaptic release sites in the presynaptic terminal (N) and the third is the average probability of transmitter release from the presynaptic release sites (P_r). Synaptic strength is defined by these parameters with Q being an expression of the postsynaptic efficacy and P_r an expression of the presynaptic efficacy.

Presynaptic modulation will alter P_r, postsynaptic modulation will alter Q and a change in the number of functional release sites, for example, a change in the number of functional synapses, will alter N. V-M analysis proposes a method whereby these parameters can be extracted by measuring synaptic amplitude fluctuations.

The experimental approach involves the acquisition of a number of records of synaptic activity at different levels of P_r. These levels can be altered by adding calcium blockers such as Cd^{2+} to the experimental solution or by altering the Ca^{2+}/Mg^{2+} ratio. The variance and mean of the individual postsynaptic current amplitudes (PSC) are measured during a stable recording period in each of the different experimental solutions, and the variance is plotted against the mean (V-M plot). The general form of such plots will be parabolic, where the initial slope is related to Q, the degree of curvature is related to P_r and the amplitude is related to N. Theoretically, a visual comparison of the different curves under different experimental conditions can thus provide insight into which synaptic parameter was altered.

The complexity of the mathematical treatment of the results depends on the data. When the V-M plot is linear or when it curves and reaches a plateau but does not descend back towards the x-axis, P_r is likely to be in the low to moderate range. The plot can then be analyzed by fitting the parabolic function from equation (1) where y is the variance of the PSC and x is the mean PSC amplitude:

$$y = (ix - x^2)/N \tag{1}$$

The free parameters i and N can be used to calculate the average synaptic parameters:

$$Q_w = i/(1 + CV_I^2) \tag{2}$$

and:

$$P_{rw} = x(i/N)(1 + CV_I^2) \tag{3}$$

The w subscript indicates that P_{rw} and Q_w are weighted averages that emphasize terminals with higher release probabilities and larger postsynaptic amplitudes (Reid and Clements, 1999). The lower limit of the number of independent release sites is given by N. CV_I is the coefficient of variation of the PSC amplitude at an individual release site and is determined experimentally (see Reid and Clements, 1999). CV is defined as:

$$CV = SD/mean \tag{4}$$

Another way to think of CV is as the inverse of the signal-to-noise ratio. A small value for CV implies good signal-to-noise.

This parameter appears in two different ways in synaptic fluctuation analysis. The first is as an indication of the variability of the response to a single vesicle (quantal variability). This can be split into two main sources-the variability at a single release site (CV_i), and the variability from site-to-site (CV_{ii}). The other way CV is sometimes mentioned in synaptic fluctuation analysis is as the CV of the evoked synaptic amplitude. It usually appears as:

$$1/CV^2$$

The reason for calculating this can be seen if this is expanded using equation (4):

$$1/(SD/mean)^2 = mean^2/SD^2 = meanx(mean/Var)$$

The initial slope of the V-M plot is the quantal amplitude Q, assuming low P_r. Therefore:

$$1/CV^2 = mean/Q$$

If synaptic modulation is by a postsynaptic mechanism then both mean and Q will be scaled by the same amount and will be unchanged. Clampfit can calculate the synaptic parameters described in equations (2) and (3) and write the results to the Lab Book.

If the V-M plot is approximately linear then P_{rw} is low and the plot can be analyzed by fitting a straight-line equation:

$$y = ix \tag{5}$$

This permits an estimate of Q_W from equation (2) but P_{rW} and N (equation 3) cannot be estimated.

Rundown Correction

Amplitude rundown is characterized by a progressive and steady decline in event amplitude. To correct for such rundown for V-M analysis, linear regression is applied to the amplitude data to obtain the relation:

$$A = mt + b$$

where A is the expected amplitude at time t (the time of the event relative to the start of the data record), m is the slope and b is the y-intercept. The data are then corrected for rundown by applying:

$$a' = a - mt$$

to each data point, where a is the observed amplitude and a' is the corrected amplitude.

As rundown correction necessarily alters the observed data, it should be applied only in those cases where the amplitude decline is noticeably linear.

BURST ANALYSIS

Poisson Surprise

The “Poisson Surprise” method (Legéndy & Salcman, 1985) is a statistical technique for detecting bursts. The name derives from the definition, being the degree to which the burst surprises one who expects a train of events to be a Poisson process. The degree of variation from a Poisson process is used as a measure of the probability that the observed train is a burst.

The Poisson Surprise (S) requires an evaluation of how improbable it is that the burst is a chance occurrence. It is computed, for any given burst that contains n events in a time interval T, as:

$$S = -log P$$

where P is the probability that, in a random train of events having the same mean rate r as the train of interest, a given time interval of length T contains n or more events. Assuming the random train of events is a Poisson process, P is given by Poisson’s formula, where:

$$P = e^{-rT} \sum_{i=n}^{\infty} (rT)^i / i!$$

Poisson Surprise Burst Analysis

In order to detect epochs of increased activity, a criterion is required for choosing the first and last events of what is to be a burst. For consistency with the definition of event rate, the number of events denoted by n in the equation above includes the first and excludes the last event.

Analysis begins with an initial pass through the data to compute the mean rate of the recorded events. In the second pass, the algorithm advances through the data, event by event, until it finds several closely spaced events, which are then taken to be a putative burst. The number of such events (N_B) can be specified, but the minimum number is 3. The interval (*I*), which is used to accept consecutive events, can either be specified or allowed to be set automatically. If automatically assigned, the minimum inter-event interval is one-half the mean inter-event interval of the entire dataset.

Once N_B events have been detected, the algorithm then tests whether including an additional event at the end of the train increases the Poisson Surprise of the burst. If so, the additional event is included. This process is repeated until a relatively long interval (greater than *I*) is encountered or the inclusion of up to 10 additional events fails to increase the Poisson Surprise. If automatically assigned, the rejection interval is set to twice the mean interval.

When the putative burst is terminated by either of the above criteria, the algorithm tests whether removing one or more events from the beginning of the burst further increases the Poisson Surprise.

Classification of bursts is accomplished by evaluating the Poisson Surprise value, which is reported with the parameters of each burst. Legéndy and Salcman classified bursts having *S* values of 10 and above as significant. For further details of statistical analysis of *S* values, the user is referred to the original reference.

Specified Interval Burst Analysis

Specified interval burst analysis uses a “burst delimiting” interval to detect and construct bursts. During analysis, this delimiting interval is compared to the “inter-event” interval. In the case of single channel data, the inter-event interval is defined as the beginning of one event to the beginning of the next. In the case of peak time data, the inter-event interval is the time between event peaks.

The algorithm examines each inter-event interval. If this interval is less than or equal to the specified delimiting interval then a putative burst is assumed. If the subsequent inter-event intervals are less than or equal to the delimiting interval, those events are added to the burst. The burst is ended if an inter-event interval greater than the delimiting interval is encountered. If the number of events in the burst is equal to or greater than a specified minimum number of events then the burst is accepted.

PERI-EVENT ANALYSIS

Peri-event analysis consists of the measurement of the time of occurrence of events over specified intervals relative to a “central” event (e.g. a stimulus-evoked action potential). This generates data of the form $T_b < T_c < T_a$ where T_b and T_a are the times of occurrence of events before and after the central event, respectively. T_c is the time of the central event, which is always set to zero and T_b and T_aa are adjusted accordingly. Thus, if T_{co} is the original, nonzero time of the central event, and T_{bo} and T_{co} are the original times of the events before and after T_{co}, for the time of occurrence of each event within the specified ranges before and after T_{co} we have:

$$T_c = T_{co} - T_{co} \quad (= 0)$$

$$T_b = T_{bo} - T_{co}$$

$$T_a = T_{ao} - T_{co}$$

These data can be displayed as a raster plot, where the data from each trace are displayed as individual points. A composite histogram having a specified bin width, using the times T_b and T_a from all of the data traces, can also be generated. T_c is at $x = 0$ in both types of graph.

Peri-event analysis is optimized to use event detection data that have been generated by Clampfit. The central events must be “tagged” explicitly during event detection. This analysis cannot otherwise detect T_c. Data from other sources can also be used so long as peak time data and tagged events exist. This might require manual tagging of the results (a tag is designated by “T”), which would require setting up a separate tag column.

For general event detection, the times of occurrence of the events is the peak time relative to the start of the trace. For single-channel data the time of occurrence is the start time of each opening relative to the start of the trace.

P$_{(OPEN)}$

The probability of a channel being open (P_{open}) provides an indication of the level of activity of an ion channel. In the simplest case, it is given by:

$$P_{open} = \frac{t_o}{T}$$

where t_o is the total time that the channel was found in the open state and T is the total observation time. This equation holds true when there is a single channel in the patch and, consequently, no multiple openings in the data record. If a patch contains more than one of the same type of channel then:

$$P_{open} = \frac{t_o}{NT}$$

where N is the number of channels in the patch, and:

$$T_o = \sum Lt_o$$

where L is the level of the channel opening. This assumes that a level 2 opening, for example, is the result of two superimposed level 1 openings of the same type of channel, where a level 1 opening is assumed to be the opening of a single channel. If different levels are composed of different types of channels then this calculation no longer holds true.

In general, N should be assigned a value equal to or greater than the number of levels in the single-channel record. It cannot be assumed that N is equal to the number of levels because N channels in a patch can generate fewer than N levels if N channel openings never overlapped during the course of recording. This can easily be the case if the channel activity is low and/or the openings are brief. However, the minimum number of channels in the patch is at least the number of observed levels.

If the number of channels is not known and there is reason to suspect that the number of levels does not accurately reflect the number of channels, the probability of the channel being open, referred to as NP_o, is be computed by:

$$NP_o = \frac{T_o}{T_o + T_c}$$

where T_c is the total closed time.

11. Curve Fitting

INTRODUCTION

Clampfit offers a powerful and flexible tool for fitting curves to data. An extensive set of options should satisfy the most demanding requirements. Four different search methods support a large number of predefined functions as well as user-defined (custom) functions. Automatic seeding of function "parameters" is provided for the predefined functions. Graphically assisted seeding is also available for more demanding fits that require accurate determination of initial seed values.

A "fitting method" is composed of a search method, a minimization method and an optional weighting method. The search method is the algorithm that reduces the difference between the data and the fitted function. The minimization method is used to specify the quantity that is to be minimized (or maximized in the case of maximum likelihood).

The search methods include Levenberg-Marquardt, variable metric, Simplex and Chebyshev.

The minimization methods include sum of squared errors, maximum likelihood, mean absolute and minimax.

The weighting methods include function weighting, data weighting, bin width weighting or no weighting.

Linear and polynomial regression routines are also provided. These non-iterative methods are automatically used when linear regression or polynomial regression is selected from the predefined function list. However, custom-defined linear or polynomial functions can only be fitted by means of one of the iterative search methods.

Fitting Model

A "fitting model", or simply, "model" is defined as any function that is fitted to the data. Functions with different numbers of terms are considered to be different models. For example, a two-term exponential function and a three-term exponential function represent different models.

Fitting to Split-Clock Data

Some older data might have been acquired using two different sampling rates (split clock). Most predefined functions and all custom functions can be fitted to such data since the fitting deals with X-Y data pairs and is unaffected by the degree of uniformity of spacing along the X axis. Exceptions include those predefined functions that require the X axis to be either normalized or converted to integers, and functions being fitted with the Chebyshev search method, where uniform spacing along the X axis is required.

Note that, for custom functions, it is up to you to ensure that the function deals with split clock data properly. If the custom function does demand uniform data spacing along the X axis then the fit to split-clock data will most likely be compromised.

Note that dual sample-interval data can be converted to a single sample interval by using Analyze > Interpolation, as described in the *"Evaluation of Multicomponent Signals: Sensillar Potentials with Superimposed Action Potentials"* on page 116 of Chapter 7, *"Clampfit Tutorials"*.

Function Parameters

The term "parameters" refers to those coefficients of the fitting function that are adjusted during fitting. For example, in the following function, A, τ and C are the function parameters. In all predefined functions the variable C is a constant offset in the y direction:

$$f(x) = Ae^{-t/\tau} + C$$

Parameters are or are not adjusted by the fitting routine depending on whether they are "free" or "fixed". Fixed parameters, which are essentially constants, are always assigned a standard error of zero.

Parameter Errors

All search methods report a standard error for each parameter in the fitting function. Parameter errors are estimated by evaluation of a covariance matrix using Gauss-Jordan elimination. The method of error evaluation is identical for all fitting methods, so the results of fitting by different methods can be compared directly.

In some cases parameter errors cannot be estimated because the covariance matrix cannot be evaluated. In this unlikely event the message "Could not compute parameter errors." is given in the Results window.

Note that the parameter errors provide an estimate of the uncertainty in the determination of the parameters themselves. They do not necessarily provide information about the goodness of the fit. The correlation coefficient and the standard deviation of the fit are more reliable indicators of the quality of the fit. In fact, if the fit is poor the parameter errors are likely to be meaningless. In other words, the parameter errors are an indication of how reliably the parameters of a given model were determined for a particular dataset, where small errors suggest that the parameter estimates are reliable regardless of the quality of the fit. Therefore, the parameter errors can be quite small although the deviation between the fitted curve and the data might be quite large.

Alternatively, for small datasets the parameter error estimates can be quite large (perhaps as large or even larger than the parameter estimates themselves), but the fit, nevertheless, can still be quite good. Clearly, statistical parameters such as estimated errors cannot be as reliable with small datasets as with larger sets.

Fitting Failure

The fitting algorithms in Clampfit are very robust and will perform satisfactorily in most cases. However, there is always the possibility that a fit will fail. The most likely reasons for a fit failure are:

- The data are very poorly described by the fitting function.
- The initial seed values were very inaccurate.
- Sign restrictions were applied and the search algorithm cannot assign acceptable positive values to the function parameters, i.e. the data cannot be reasonably well described by the sign-restricted fitting function.

In the event of a fit failure possible solutions are:

- Ensure that the data are indeed reasonably well-represented by the fitting function. If not, select or define a different fitting function. Also, try a different number of terms or run a model comparison.
- Assign more accurate seeds to the fitting function. Graphical seeding should be very helpful for this.
- Use the variable metric search method. This search method is the most reliable for forcing function parameters positive.
- Disable the Force parameters positive option.
- Reduce the tolerance. This could result in a poorer, although still acceptable, fit.
- Reduce the maximum number of iterations. Sometimes an acceptable fit can be achieved (as judged by the parameter errors and the quality of the fitted curve) even though the fit does not converge. This is especially true for Simplex, which can continue to search for many iterations even though it is very close to the function minimum.

In the event of a failed fit the error is reported in the Results window. When fitting multiple sweeps, errors do not cause execution to stop. If an error occurs while fitting a given sweep, the error is recorded in the Results window and fitting continues with the next sweep. Therefore, if you have fitted a series of sweeps you should check the fitting results to ensure that all sweeps have been successfully fitted.

Numerical Limitations

- The maximum number of data points that can be fitted is 110,000.
- The maximum number of function terms (where applicable) is 6.
- The maximum power (where applicable) is 6.
- The maximum number of parameters in a custom function is 24.
- The maximum number of independent variables in a custom function is 6.
- Only one dependent variable is allowed in a custom function.
- The maximum number of points for functions that contain a factorial term is 170.

Units

The fitting routines in Clampfit do not make any assumptions about the units of the data. The variety of data sources and the potential for various data transformations makes the automatic tracking or assignment of units virtually impossible. Consequently, units are not displayed along with the reported parameter values. It is up to the user to determine the units of a particular function parameter.

THE LEVENBERG-MARQUARDT METHOD

The Levenberg-Marquardt method supports the least squares, mean absolute and minimax minimization methods. The explanation given here is for least squares minimization but the general principle is the same for all minimization functions.

The sum of squared errors (SSE) is first evaluated from initial estimates (seed values) for the function parameters. A new set of parameters is then determined by computing a change vector P that is added to the old parameter values and the function is reevaluated. The value of P will depend on the local curvature in the "parameter space" that can be evaluated to determine the optimal rate and direction of descent toward the function minimum. This process continues until the SSE is "minimized" at which time the fit is said to have converged. The criteria by which is judged to be at its minimum are different for the different search methods.

The Levenberg-Marquardt search method combines the properties of the steepest descent and the Gauss-Newton methods. This is accomplished by adding a constant λ to the diagonal elements of the Hessian matrix that is associated with the gradient on the parameter space. If λ is large the search algorithm approaches the method of steepest descent. When λ is small the algorithm approaches the Gauss-Newton direction.

The method of steepest descent can optimally find major changes in the data and thus works best in the early stages of the fit when the residual of the sum of squares is changing substantially with each iteration. The Gauss-Newton method is best for smoothing out the fit in later stages when these residuals are no longer changing substantially (see Schreiner *et al.* 1985). The Levenberg-Marquardt method requires that the first derivative of the function $f(x,P)$ be evaluated with respect to each parameter for each data point. These derivatives are used to evaluate the "curvature" in the local parameter space in order to move in the direction of the perceived minimum. For predefined functions the exact derivative is calculated. For custom functions a numerical derivative (central difference) is computed using a step size of 10^{-7}.

Note that as the fit progresses some steps may result in a poorer (larger) value of the SSE. However, the general trend is a reduction in the SSE.

The Levenberg-Marquardt method does not report SSE during the fitting process but rather reports the standard deviation (σ). However, σ follows the same trend as the SSE, that is, if the SSE increases then σ also increases, and vice-versa. In fact, the standard deviation is reported by all search methods, providing a standard criterion for judging the fitting quality regardless of the search method. The standard deviation is given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(Obs - Exp)^2}{n-1}}$$

where n is the number of data points, Obs is the observed value and Exp is the expected value as calculated using the fitting function.

Levenberg-Marquardt Convergence

Convergence is reached when the parameter change vectors go to zero, which occurs when a minimum is reached in the local parameter space. Note that because of this the fitting function might converge to a minimum that is not necessarily the lowest (global) minimum in the entire parameter response surface. Convergence to a local minimum often results in a poorly fitted curve and so is easily recognized. If you suspect that the fit has converged on a local minimum, you should specify new fitting seed values (graphically-assisted seeding is very useful here) and retry the fit. Alternatively, use a different fitting method. For example, the Simplex search method is not as prone to convergence at a local minimum.

The iterations are also stopped (convergence is assumed) when the change in the minimization function (for example, the SSE) is less than a preset value. This value can be set in the Precision field in the Function>Method tab of the fitting dialog. The default value is 10^{-6}.

Normally, it is preferable to allow the parameters to converge naturally. Convergence on a precision criterion can result in a poorer fit especially if the precision criterion is reached before the individual parameters have converged. On the other hand, some “difficult” fits might require hundreds or even thousands of iterations if only the change vector criterion is used for convergence. In order to favor convergence on the basis of change vectors but to also allow difficult fits to converge on the basis of an acceptable “precision” value, the fitting routine converges on the precision criterion only if this criterion has been met over at least 100 successive iterations. Given this criterion it is not likely that further improvements in the minimization function will lead to a better fit, so the iterations will stop.

Levenberg-Marquardt Precision

The default Levenberg-Marquardt precision is 10^{-6}.

The Levenberg-Marquardt precision sets the minimum absolute change in the minimization function (for example, the SSE) that signifies convergence. This minimum difference must be satisfied over at least 100 successive iterations. That is, if:

$$absolute(SSE(old) - SSE(new)) < Precision$$

over 100 consecutive iterations, convergence is assumed. A less stringent precision value could facilitate convergence for a particularly difficult dataset, but often at the expense of fitting accuracy.   In any case, the statistics of the fit should always be carefully evaluated in order to determine whether or not the fit is acceptable.

THE SIMPLEX METHOD

The Simplex method supports the least squares, mean absolute, maximum likelihood and minimax minimization methods. The explanation given here is for least squares minimization but the general principal is the same for all minimization functions.

The Simplex search method is based on the algorithm of Nedler and Mead (1965), and is an example of a direct search approach that relies only on the values of the function parameters. It does not consider either the rate or the direction by which the function is approaching the minimum on the parameter response surface. However, the direction in which the function parameters proceed is not purely random but rather relies on a clever strategy that takes advantage of the geometry of the response surface.

A simplex is a geometric figure that has one more dimension than the parameter space in which it is defined. The vertexes of the simplex are first separated by adding an "offset" to the initial seed values. The function to be minimized is then evaluated at each vertex to identify the lowest and highest response values. For example, a simplex on a two-dimensional space (corresponding to a two-parameter function) is a triangle that may have the following appearance:

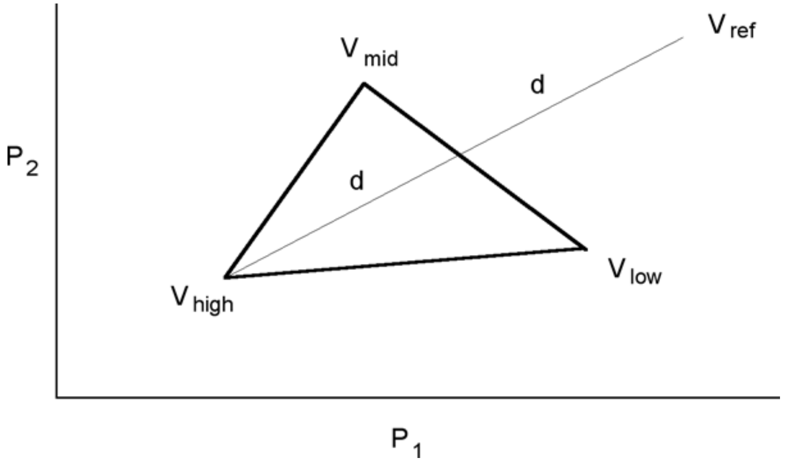


Figure 11.1: A simplex for a two-parameter function.

where P_1 and P_2 are the parameters, V_{high} is the vertex which has the highest (worst) function value, V_{low} is the vertex which has the lowest (best) function value and V_{mid} represents an intermediate function value. A "downhill" direction is established by drawing a line from V_{high} through a point midway between V_{mid} and V_{low}. The algorithm then tries to find a point along this line that results in an SSE which is lower than the existing vertexes.

The simplex changes shape by reflection, contraction, expansion or shrinkage. The first point tested is the reflected point V_{ref} which lies a distance of 2d along the line from V_{high}. This reflected simplex is accepted if its response is neither worse than V_{high} nor better than V_{low}. If the response is better than V_{low} then the algorithm performs an expansion by moving a distance of 4d along the line from V_{high}. The expansion is accepted if it has a lower (better) response than the previous best. Otherwise the reflection is accepted.

If the reflection results in a higher (worse) response than V_{high} then the algorithm tests a contraction by moving a distance of 0.5d toward the midpoint on the line. If this produces a better response then the simplex is accepted; otherwise shrinkage occurs where all vertexes except V_{low} move toward the midpoint by one-half of their original distance from it.

The advantage of the Simplex algorithm is that it is considerably less sensitive to the initial seed values than the gradient search algorithms. It will rapidly approach a minimum on the parameter response surface usually in the space of several tens of iterations for a multicomponent function.

The disadvantage of the Simplex fitting method is that its sensitivity does not increase when it is in the vicinity of a minimum on the parameter space.

Another problem can arise in that the Simplex algorithm may find what it perceives to be a local minimum but the fractional error is still greater than the convergence criterion (see below). In this case iterations may continue endlessly. To circumvent this problem, the fitting routine will stop when there is no change in the fractional error for 30 iterations. Even if the above error criterion is not met, the fit is assumed to have converged. This occurrence is reported as a "Stable Fractional Error" in the Results Window. Note that the displayed value of σ may be the same for many more than 30 iterations before the fitting routine is terminated. This is because σ is reported as a single precision value whereas the fractional error is a double precision value.

Note that the weighting options are not available with Simplex fitting. This is because weighting interferes with the "travel" of the simplex and greatly reduces the chances of convergence.

Simplex Convergence

The Simplex algorithm can converge in one of three ways. The iterations are stopped when the fractional error is less than or equal to a preestablished "precision" value.

The simplex is moved over the parameter space until the ratio of the response of the best and worst vertexes reaches a preset minimum fractional error, at which point the function is said to have converged:

$$Fractional\ Error\ = \frac{V_{high} - V_{low}}{V_{high}}$$

The quantities on the right-hand side of the equation are based on one of four minimization methods that Simplex can use, namely least squares, maximum likelihood, mean absolute or minimax.

The fractional error is computed for the simplex for each dimension (each having a V_{high} and V_{low} simplex). All simplexes must have a fractional error less than the value of precision for convergence.

Simplex Precision

The default Simplex precision is 10^{-5}.

The Simplex search is deemed to have converged when the fractional error is less than or equal to the Precision value. The fractional error can be based on one of four minimization methods, namely least squares, maximum likelihood, mean absolute or minimax.

THE VARIABLE METRIC METHOD

The variable metric method supports the least squares and maximum likelihood minimization methods only.

Variable metric algorithms are designed to find either the minimum or the maximum of a multi-dimensional non-linear function *f*. The minimum is used in chi-squared or least squares applications and the maximum in maximum likelihood estimation.

Clampfit typically uses minimization of least squares, which is asymptotically equivalent to likelihood maximization for most applications (Rao, 1973). The parameter values that determine the global minimum of f are optimal estimates of these parameters. The goal of the variable metric algorithm is to find these estimates rapidly and robustly by an iterative method.

Clampfit uses the variable metric algorithm introduced by Powell (1978) and implemented by Dr. Kenneth Lange (UCLA). At each iteration the algorithm computes the exact partial derivative of *f* with respect to each parameter. The values of these derivatives are used in building an approximation of the Hessian matrix, which is the second partial derivative matrix of *f*. The inverse of the Hessian matrix is then used in determining the parameter values for the subsequent iteration.

Variable metric algorithms have several desirable characteristics when compared with other methods of non-linear minimization. Like the simplex algorithm, variable metric algorithms are quite robust, meaning that they are adept at finding the global minimum of f even when using poor initial guesses for the parameters. Unlike simplex, however, convergence is very rapid near the global minimum.

Variable Metric Convergence

The variable metric search method converges when the square of the residuals, or the maximum likelihood estimate if using likelihood maximization (see *"Maximum Likelihood Estimation"* on page 201), does not change by more than a preset value over at least four iterations. This value can be set in the Precision field in the Function > Method tab of the Fitting dialog. The default value is 10^{-4}.

Variable Metric Precision

The default variable metric precision is 10^{-4}.

The variable metric search is assumed to have converged when the square of the residuals, or the maximum likelihood estimate if using likelihood maximization, does not change by more than the Precision value over at least four iterations.

THE CHEBYSHEV TRANSFORM

The Chebyshev technique is an extremely rapid, stable, noniterative fitting technique with a goodness of fit comparable to that of iterative techniques. It was developed by George C. Malachowski and licensed to Axon Instruments. The explanation in this section only describes how the Chebyshev transform is used to fit sums of exponentials.

The Chebyshev Transform transforms a dataset to the Chebyshev domain (in this case, domain refers to a functional basis, in which datasets are considered in terms of scaled sums of functions), using a method equivalent to transforming data to the frequency domain using Fourier transforms. Instead of representing the data using a sum of sines and cosines, the Chebyshev transform uses a sum of the discrete set of Chebyshev polynomials (described below).

Transforming the data allows it to be fitted with various functions of biological interest: sum of exponentials, Boltzmann distribution and the power expression (an exponential plus a constant raised to a power). For each of these functions, it is possible to derive an exact, linear, mathematical relationship between the fit parameters and the coefficients of the transformed data. If the dataset has noise present, as is almost always the case, these relationships provide estimates of the fit parameters. However, as the relationships are linear, high-speed regression techniques can be applied to find the parameters.

This method has the following properties: extremely fast fitting at a rate that is independent of noise, comparable goodness of fit to that of iterative techniques, and it always finds a fit.

At present this technique has one limitation: it can only be used on datasets that have equally spaced data points. This makes it inappropriate at present for fitting histogram data with variable bin widths.

The Orthogonal Polynomial Set

An N^th order polynomial is a function of the form:

$$P_N(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots + a_Nx^N$$

where a_0, a_1, a_2, ... a_N are the coefficients of the polynomial, and a_N cannot be zero unless N is zero.

A set of polynomials consists of one polynomial from each order ($P_0(x)$, $P_1(x)$, $P_2(x)$, $P_3(x)$, ...), in which adjacent members of the set are related by a generating equation. Mathematicians usually represent a polynomial set with a single letter. In the case of the Chebyshev polynomials, the letter is "T" (from previous anglicization of Chebyshev as Tchebysheff).

A polynomial set is said to be orthogonal if every member in the set is orthogonal to every other member. In the case of the Chebyshev polynomials, T_0 is orthogonal to T_1, T_2, T_3, T_4, and so on, T_1 is orthogonal to T_2, T_3, T_4, T_5, and so on.

Orthogonal means *at a right angle*; synonyms are *perpendicular* and *normal*. Those familiar with vectors may recall that two vectors are tested for orthogonality using the dot product; two vectors are said to be orthogonal if their dot product equals zero. Similarly, two functions that are defined only at discrete points in time are said to be orthogonal if the sum of their product over all sampled points is zero. (This relation may only be true on a restricted range (e.g. $x \in [-1, 1]$), and with a weighting function present.)

Transforming Data to the Chebyshev Domain

All orthogonal function sets have the property of being able to represent a function in terms of a sum (or integral) of the members of the set. Depending on the function being represented, an appropriate scaling factor is chosen for each member of the set. A function $f(t)$ can be represented as a sum over the Chebyshev polynomials by the relation:

$$f(t) = \sum_{j=0}^{\infty} d_j T_j(t)$$

where $T_j(t)$ is the j$^{\text{th}}$ member of the Chebyshev polynomial set and d_j is its scaling factor. In general, a sum of an infinite number of Chebyshev polynomials is required to represent a continuous function. In the case of a sampled dataset with N sampled points, a sum of only N Chebyshev polynomials, from order 0 to N–1, is required to exactly represent the dataset. In this case, t is not continuous, but is rather a set of data points t_j, where $_j$ runs from 0 to N–1. The above equation then becomes:

$$f(t_i) = \sum_{j=0}^{N-1} d_j T_j(t_i). \qquad \text{for } i = 0, \ldots, N-1 \qquad (1)$$

This sum of polynomials exactly equals the function at *all* points in time, even though the individual members may only cross the function at a few points.

A function that has been represented this way is said to have been *transformed* to the Chebyshev domain, and the scaling factors (the d_js) are usually referred to as the coefficients of the transform. Do not confuse these transform coefficients with the individual coefficients that make up each of the member Chebyshev polynomials! The member polynomials' coefficients never change; the coefficients of the transform change with $f(tj)$.

Calculating the Coefficients

The orthogonality property of the Chebyshev polynomials makes calculating the d_js straightforward. Recall that every member T_k is orthogonal to every other member T_j, for all $k \neq j$. To determine each coefficient (d_k), both sides of the above equation are multiplied by T_k, then summed over all values of t_i:

$$\sum_{i=0}^{N-1} T_k(t_i)f(t_i) = \sum_{i=0}^{N-1} T_k(t_i) \sum_{j=0}^{N-1} d_j T_j(t_i)$$

where T_k is the member whose coefficient is to be determined. Rearranging, and summing over all data points t_i eliminates all T_js except T_k, leaving (after several steps):

$$\sum_{i=0}^{N-1} T_k(t_i)f(t_i) = d_k \sum_{i=0}^{N-1} T_k^2(t_i)$$

The summation on the right-hand side is usually written as a normalization factor (R_k). Solving for d_k:

$$d_k = \sum_{i=0}^{N-1} T_k(t_i)f(t_i)/R_k \qquad (2)$$

The Discrete Chebyshev Polynomials

The generating equation for the Chebyshev polynomials is given by (Abramowitz and Stegun, p. 792):

$$j(N-j) \cdot T_j(t) = (2j-1)(N-1-2t) \cdot T_{j-1}(t) - (j-1)(N-1+j) \cdot T_{j-2}(t)$$

where $T_j(t)$ is the Chebyshev polynomial being generated, and N is the number of data points. It is clear that each $T_j(t)$ depends on the previous two members in the set: $T_{j-1}(t)$ and $T_{j-2}(t)$. The zeroth member of this set is defined to be: $T_0 = 1$, from which all higher-order members may be derived. T_1 and T_2 are shown below:

$$T_1(t) = 1 - \frac{2}{(N-1)}t$$

$$T_2(t) = 1 - \frac{6}{(N-2)}t + \frac{6}{(N-1)(N-2)}t^2$$

Clearly, T_0 is a horizontal line, T_1 is a sloping line and T_2 is a parabola.

Isolating the Offset

The offset of a dataset can be isolated from the data in the Chebyshev domain. Consider the general case of a function containing an offset; we can rewrite this function in terms of its non-constant and its constant parts:

$$f(t_i) = g(t_i) + K$$

where $g(t_i)$ is the non-constant part of the function. Using this function in Equation 2 we derive:

$$d_j = \left(\sum_{i=0}^{N-1} T_j t_i g t_i / R_j\right) + \left(\sum_{i=0}^{N-1} T_j t_i K / R_j\right)$$

The above equation is simply the sum of the Chebyshev transforms of g and K. This can be seen if we write the transform coefficients of g as $d'_j(g)$, and the transform coefficients of K as $d''_j(K)$:

$$d_j(g + K) = d'_j(g) + d''_j(K)$$

However, $T_0 = 1$ implies that the Chebyshev transform of K is nonzero only for the zeroth coefficient ($d''_0(K)$). (K can be rewritten as KT_0, and T_0 is orthogonal to all other Chebyshev polynomials.) We therefore have:

$$d_0 = d'_0(g) + d''_0(K)$$

$$d_j = d'_j(g) \qquad \text{for } j>0$$

Once a dataset has been transformed to the Chebyshev domain, we can isolate the effect of the constant offset by not using d_0 in our calculations of the other parameters.

Transforming an Exponential Dataset to the Chebyshev Domain

Suppose we wish to transform an exponentially decaying signal $f(t_i)$ to the Chebyshev domain, where $f(t_i)$ is defined as:

$$f(t_i) = a_0 + a_1 e^{-t_i/\tau}$$

To exactly represent $f(t_i)$ requires N coefficients, where N is the number of data points in the set. However, we can approximate f quite well with a sum of just a few polynomials. A pure, noiseless exponential with time constant 25 ms, sampled every 1 ms from t = 0 to t = 255 ms, when transformed to the Chebyshev domain has the following first 11 coefficients:

Chebyshev transform of

d_0	0.0996188372
d_1	0.239563867
d_2	0.260135918
d_3	0.196609303
d_4	0.115307353
d_5	0.0552865378
d_6	0.0223827511
d_7	0.00782654434
d_8	0.00240423390
d_9	0.000657552096
d_{10}	0.000161851377

The magnitudes of the coefficients rise to a peak at d_2, then decline slowly as the coefficient index increases. Using these coefficients to approximate the exponentials by forming a sum of the first 11 Chebyshev polynomials approximates the original exponential to *at worst* 4.5 x 10^{-5}. Below is a graph of the exponential data, with the above Chebyshev fit superimposed upon it. Inset in this graph is a greatly magnified view of the residuals between the fit and the exponential. Note the polynomial-like oscillation of the residuals about the data.

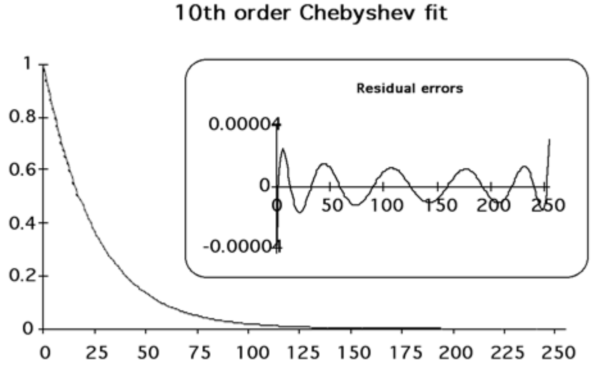


Figure 11.2: Graph of exponential data with Chebyshev fit superimposed.

Although 11 Chebyshev coefficients are sufficient to well approximate an exponential dataset, the presence of high frequency noise would require many more coefficients to accurately represent the data. If fewer coefficients are used, the approximation will appear filtered.

Integrating an Exponential Function in the Chebyshev Domain

Let us say that we have a function *f* (*t*) that we wish to integrate, and that its integral is *F* (*t*). In the discrete domain, where we only have a finite set of data points (usually evenly spaced), we write the discrete integral as:

$$\sum_{0}^{t-1} f(t')dt' = F(t)$$

The discrete integral *F* (*t*) is defined at each value *t* by the sum of the previous values from 0 to *t*–1. Using *t*–1 as the end point of the integration serves to ensure that the forward difference equation is equal to *f* (*t*):

$$F(t+1) - F(t) = f(t)$$

A *difference equation* in the discrete domain is analogous to a differential equation in the continuous domain. The equation shown above in the continuous domain would be expressed as $dF/dt = f(t)$. Forward difference refers to using *t* and *t*+1 to form the difference.

How are the Chebyshev transforms of the integral F and f related? If we were to transform f to the Chebyshev domain, we would obtain a set of coefficients $d_j\,(f)$. Similarly, if we were to transform F, we would obtain a different set of coefficients $d_j\,(F)$. Comparing these two sets of coefficients, $d_j\,(f)$ and $d_j\,(F)$ we would find the relation:

$$D_j(F) = \frac{1}{2}\left(\frac{(N+j+1)}{2j+3}d_{j+1}(f) - d_j(f) - \frac{N-j}{2j-1}d_{j-1}(f)\right) \quad \text{for all } j > 1 \qquad (3)$$

where $d_j\,(F)$ is the j^{th} coefficient of $F(t)$ and N is the number of points in the data. Note that this equation cannot tell us the value of $D_0(F)$, as there is no $d_{-1}\,(f)$ coefficient.

(This formula is derived in *A Method that Linearizes the Fitting of Exponentials*, G. C. Malachowski). This equation is critical to the use of this technique. Proof of this relation is long; those interested may refer to the appendix of the above paper. Briefly though, it can be described as follows: integrating the Chebyshev transform of a function is the same as the sum of the integrals of each of the Chebyshev polynomials making up the transformation. The integral of a polynomial is itself a polynomial. It turns out that after much simplification and rearrangement, each coefficient in the transform of the integral is a sum of the two adjacent coefficients in the transform of the original function.)

If f is an exponential function, the following, very similar, relationship exists:

$$\frac{d_j(f)}{k} = \frac{1}{2}\left(\frac{(N+j+1)}{2j+3}d_{j+1}(f) - d_j(f) - \frac{N-j}{2j-1}d_{j-1}(f)\right) \quad \text{for all } j > 1 \qquad (4)$$

where k is defined as:

$$k = e^{-1/\tau} - 1$$

or, solving for τ:

$$\tau = \frac{-1}{log_e(k+1)} \qquad (5)$$

Basically, these equations tell us that any adjacent triplet of Chebyshev coefficients forms an exact relationship that tells us the value of tau. Note how Equation 4 further restricts the value of j to be greater than one: it is only true for those coefficients that do not contain the constant offset term.

Thus, integrating an exponential function in the Chebyshev domain allowed us to determine the value of tau. This is similar to the case of integrating an exponential function in the continuous domain:

$$\int e^{-t/\tau}dt = -\tau e^{-t/\tau} + C$$

For reasons that will become clear in the section on the fitting of two exponentials, the right-hand side of Equation 4 can be written as $d1_j\,(f)$, which stands for the Chebyshev coefficients of the first integral of f. Equation 4 then becomes:

$$\frac{d_j(f)}{k} = d1_j(f) \qquad \text{for } j > 1 \qquad (6)$$

where k is as defined in Equation 5.

Calculating Tau

Now we can calculate τ using Equations 4 and 5. Choose any triplet of Chebyshev coefficients, and use those values in Equation 4 to get the value of k. Then use k in Equation 5 to calculate τ. Every triplet has the same, redundant information built into it: triplet d_1, d_2, d_3, triplet d_2, d_3, d_4, and so on. The following is an example of the values of τ predicted using the 8 triplets of the first eleven Chebyshev coefficients.

Note: remember that d_0 cannot be used as it contains information about the offset of the exponential, if present.

Chebyshev Calculations of τ

d_1, d_2, d_3	25.00000003
d_2, d_3, d_4	25.00000075
d_3, d_4, d_5	24.99999915
d_4, d_5, d_6	24.9999998
d_5, d_6, d_7	25.00000096
d_6, d_7, d_8	25.00000257
d_7, d_8, d_9	25.00000464
d_8, d_9, d_{10}	24.99999885

The small differences between the calculated value and the actual value (25) are due to the limited precision of the coefficients used. In general, double precision numbers are used to calculate the fit parameters (τ, the amplitude and the offset).

Calculating the Amplitude

To determine the amplitude of the exponential, we must change directions completely, and generate an exponential dataset based on the value of tau just calculated. The generated dataset will have unity amplitude and zero offset:

$$g(t_i) = 1e^{-t_i/\tau} + 0$$

Transforming this generated set to the Chebyshev domain will give us a different set of coefficients ($d'_j(g)$). Now we can determine the value of the amplitude of our dataset by comparing the $d_j(f)$s from our dataset to those of the generated set $d'_j(g)$s. Recalling that the function that we are trying to fit is:

$$f(t_i) = a_0 + a_1 e^{-t_i/\tau}$$

we can transform this function to the Chebyshev domain as:

$$d_j(f) = \left(\sum_{i=0}^{N} a_0 T_j(t_i)/R_j\right) + a_1\left(\sum_{i=0}^{N} e^{-t_i/\tau} \cdot T_j(t_i)/R_j\right)$$

The Chebyshev coefficients for $g(ti)$ are very similar to those for $f(t_i)$:

$$d'_j(g) = \sum_{i=0}^{N} e^{-t_i/\tau} \cdot T_j(t_i)/R_j \qquad \text{for all } j$$

Comparing these two equations yields the following relationship between the Chebyshev coefficients of f and those of g:

$$a_1 = \frac{d_j(f)}{d'_j(g)} \qquad \text{for all } j > 0 \qquad (7)$$

Note that the amplitude is contained redundantly in all of the coefficients of the two transforms, excluding the zeroth coefficient. This redundancy is similar to that seen in the calculation of tau.

$$a_1 = \frac{d_1(f)}{d'_1(g)} = \frac{d_2(f)}{d'_2(g)} = \frac{d_3(f)}{d'_3(g)} = \frac{d_4(f)}{d'_4(g)} = \ldots$$

Calculating the Offset

Calculating the offset is similar to calculating the amplitude: we compare the zeroth index coefficients from the two sets of transforms:

$$a_0 = d_0(f) - a_1 d'_0(g) \qquad (8)$$

Unlike τ and a_1, the offset information is not redundantly stored in the transform coefficients.

Calculating the Fit Parameters in the Presence Of Noise

If the dataset being fit contains noise, Equations 4, 5, 7 and 8 are no longer exactly true. For example, when calculating τ, each triplet gives an estimate of τ.

Estimating τ

Recall that for **j** > 1, Equation 4 shows a relationship between *k* and each triplet of Chebyshev coefficients. In the case of noise, this relationship is not strictly true; $d_j(f)/k$ is an estimate of the right side of this equation. Equation 6 then becomes:

$$d_j(f) \cong kd1_j(f)$$

The value of *k* that minimizes the following expression will be *k´*:

$$\chi^2 = \sum_{j=1}^{n} (d_j - kd1_j^2(f))$$

where the sum generally does not include all *N* of the coefficients, but generally includes only those coefficients with a significant contribution to the transform; usually *n* is chosen to be 20. Expanding the squared term, differentiating with respect to *k*, setting the derivative to 0 and rearranging gives us our estimate of *k*:

$$k' = \sum_{j=1}^{n} d_j d1_j(f) / \sum_{j=1}^{n} d1_j^2(f) \qquad (9)$$

Once *k* has been estimated, Equation 5 will give the corresponding best estimate of τ.

Estimating a_1

A similar technique is used to calculate the best estimate of the amplitude a_1. The ratios of the coefficients of our dataset [$d_j(f)$s] to the coefficients of the pure exponential [$d´_j(g)$s] now give estimates of the amplitude:

$$\frac{d_j(f)}{d'_j(g)} \cong a_1$$

This can be rewritten as:

$$d_j(f) \cong a_1 d'_j(g)$$

As in the case of estimating tau, we form a linear regression equation to find the best estimate of a_1 ($a_1´$):

$$\chi^2 = \sum_{j=1}^{n} (d_j(f) - a_1 d'_j(g))^2$$

Expanding the squared term, differentiating with respect to a_1, setting the derivative to 0 and rearranging gives us:

$$a'_1 = \sum_{j=1}^{n} d_j(f) d'_j(g) / \sum_{j=1}^{n} (d'_j(g))^2 \qquad (10)$$

Estimating a_0

The estimate of a_0 is calculated by substituting the above value of a'_1 into Equation 8.

Fitting the Sum of Two Exponentials

In the two exponential case, two taus must be found. To do so, we shall integrate the function to be fit twice, and solve the resulting set of simultaneous equations for τ_1 and τ_2. (This procedure is somewhat complicated.) Once we have the two taus, solving for the amplitudes and the offset is a simple extension of the procedure for fitting a single exponential.

Suppose we wish to transform a signal $f(t_i)$ that is the sum of two exponentially decaying functions to the Chebyshev domain, where $f(t_i)$ is defined as:

$$f(t_i) = a_0 + a_1 e^{-t_i/\tau_1} + a_2 e^{-t_i/\tau_2}$$

Let $g_{\tau_1}(t_i)$ represent a unity-amplitude exponential with time constant τ_1 and let $g_{\tau_2}(t_i)$ represent a unity-amplitude exponential with time constant τ_2. Then we can write:

$$f(t_i) = a_0 + a_1 g_{\tau_1}(t_i) + a_2 g_{\tau_2}(t_i) \qquad (11)$$

Transforming both sides of this equation gives us:

$$d_j(f) = \left(\sum_{i=0}^{n} (a_0 + a_1 g_{\tau_1}(t_i) + a_2 g_{\tau_2}(t_i)) T_j(t_i)\right) / R_j$$

or

$$d_j(f) = d_j^{offset}(a_0) + a_1 d_j^{\tau_1}(g_{\tau_1}) + a_2 d_j^{\tau_2}(g_{\tau_2})$$

where $d_j^{offset}(a_0)$, $d_j^{\tau_1}(g_{\tau_1})$ and $d_j^{\tau_2}(g_{\tau_2})$ are the Chebyshev coefficients of a_0, g_{τ_1} and g_{τ_2} respectively. To isolate the constant term from the calculations that follow, we shall only use coefficients where $j > 0$, yielding:

$$d_j(f) = a_1 d_j^{\tau_1}(g_{\tau_1}) + a_2 d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 0 \qquad (12)$$

since the offset is only contained in the zeroth coefficients.

Using the Coefficients of *f* and its Integrals to Determine the Taus

What if we were to integrate both sides of Equation 12? Since we are dealing with discrete data points, we use a summation. From Equations 4, 5 and 6, we know the coefficients of the integral of an exponential function, and we know how those coefficients are related to the coefficients of the exponential function itself. Applying those relationships to this sum of two exponentials case yields:

$$d1_j(f) = a_1 d1_j^{\tau_1}(g_{\tau_1}) + a_2 d1_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 1 \qquad (13)$$

or using Equation 6 to rewrite in terms of the coefficients of g_{τ_1} and g_{τ_2} themselves:

$$d1_j(f) = \frac{a_1}{k_1} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2} d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 1 \tag{14}$$

where

$$\tau_1 = \frac{-1}{log_e(k_1+1)} \quad \text{and} \quad \tau_2 = \frac{-1}{log_e(k_2+1)} \tag{15a, b}$$

Integrating both sides of Equation 13 again:

$$d2_j(f) = \frac{a_1}{k_1} d1_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2} d1_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 2$$

where we write the Chebyshev coefficients of the second integral of f as $d2_j\,(f)$. Note that j now must be greater than two. (The exact reason for this is beyond the scope of this description (see Malachowski's paper) however, briefly, it is required to isolate the effect of the offset from the calculation of the taus.) Substituting in Equation 6 again gives us our final relation that we need to determine τ_1 and τ_2:

$$d2_j(f) = \frac{a_1}{k_1^2} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2^2} d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 2 \tag{16}$$

Solving a Set of Simultaneous Equations to Determine k_1 and k_2

Equations 12, 14 and 16 now form the three relations that we need in order to determine the taus. Rewriting them below we have three equations in three unknowns d_j, $d1_j$ and $d2_j$, and restricting j to be the same in all cases:

$$d_j(f) = a_1 d_j^{\tau_1}(g_{\tau_1}) + a_2 d_j^{\tau_2}(g_{\tau_2})$$

$$d1_j(f) = \frac{a_1}{k_1} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2} d_j^{\tau_2}(g_{\tau_2})$$

$$d2_j(f) = \frac{a_1}{k_1^2} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2^2} d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 2$$

In order for these three equations to be simultaneously true, there must exist a pair of parameters x_1 and x_2 such that for all $j > 2$:

$$d_j + x_1 d1_j + x_2 d2_j = 0 \tag{17}$$

The solution to this equation is a straight line in x_1–x_2 coordinate space. To solve it, we add up the three simultaneous equations, and gather the like terms to find:

$$d_j + x_1 d1_j + x_2 d2_j = a_1 d_j^{\tau_1}(g_{\tau_1})\left(1 + \frac{x_1}{k_1} + \frac{x_1}{k_1^2}\right) + a_2 d_j^{\tau_2}(g_{\tau_2})\left(1 + \frac{x_1}{k_2} + \frac{x_1}{k_2^2}\right)$$

The values of x_1 and x_2 that satisfy this equation are:

$$x_1 = -(k_1 + k_2) \qquad (18a, b)$$

$$x_2 = k_1 k_2$$

as can be seen by substituting these values into the above equation.

Our strategy will be to solve for x_1 and x_2, from them calculate the values of k_1 and k_2, and finally use Equations 15a and b to calculate the corresponding values of τ_1 and τ_2. To do so, we must first solve for k_1 and k_2 in terms of x_1 and x_2 (Equations 18a and b are the converse). There is no direct, algebraic method to do so, but we can recognize that Equations 18a and b are the roots of the quadratic polynomial:

$$k^2 + x_1 k + x_2 = 0 \qquad (19)$$

as can be seen by factoring the polynomial into the product of its roots:

$$k^2 + x_1 k + x_2 = (k - k_1)(k - k_2)$$

This means that we can determine k_1 and k_2 by using x_1 and x_2 to form the above quadratic polynomial, and then solving for its roots. For a quadratic polynomial, we use the quadratic formula. For higher-order polynomials, such as are used for fitting higher order exponentials, an iterative, root-finding method is used (Newton-Raphson iteration: see Kreyszig 1983, pp. 764–66).

What if there are not two, real roots? Recall that the quadratic formula either yields two real roots, one real root, or two complex roots. This corresponds geometrically to two crossings of the X axis, one tangential "touch" of the axis, or no crossings of the axis.

If there is one real root, then the data being fit only consisted of a single exponential. In this case, this technique would yield two taus with the same value as that of the single exponential, and with amplitudes each one-half of the amplitude of the single exponential.

If there are two complex roots, the data being fit is not a pure exponential. Rather, it is the product of an exponential and a harmonic function (e.g. a cosine). This function is commonly called a ringing response, or an exponentially-damped cosine. This can be seen by substituting a complex number into Equations 15a and b ($a \pm bi$), rewriting the resulting number in terms of a complex exponential ($re^{i\theta}$), where r is $\sqrt{(a+1)^2 b^2}$ and θ is $tan^{-1}/(b(a+1))$, taking the logarithm, substituting back into Equation 11, and simplifying.

Finding the Taus in the Presence of Noise

In the presence of noise, $f(t_j)$ is not an exact sum of exponentials, and therefore the Chebyshev coefficients d_j, $d1_j$ and $d2j$ do not lie along a straight line, but are scattered:

$$d_j + x_1 d1_j + x_2 d2_j \cong 0$$

To find the best line through the data, we form the following regression equation, and minimize the χ^2 value:

$$\chi^2 = \sum_{j=2}^{n} (d_j - x_1 d1_j - x_2 d2_j)^2 \qquad (20)$$

The best values for x_1 and x_2 are determined by expanding this relation, minimizing it first with respect to x_1, then with respect to x_1. After rearranging we have the following set of simultaneous equations:

$$\begin{aligned} \sum_{j=2}^{n} d_j d1_j &= x_1 \sum_{j=2}^{n} d1_j^2 + x_2 \sum_{j=2}^{n} d1_j d2_j \\ \sum_{j=2}^{n} d_j d2_j &= x_1 \sum_{j=2}^{n} d1_j d2_j + x_2 \sum_{j=2}^{n} d2_j^2 \end{aligned} \qquad (21a, 21b)$$

Direct solution of simultaneous equations is a well-known problem in mathematics; an iterative matrix technique is used here (Gauss-Seidel iteration, in Kreyszig, pp. 810–12. This technique has extremely fast convergence, particularly in the sum of m exponentials case). This allows us to easily solve for the more difficult case of finding the solution to a set of m simultaneous equations, which must be used when fitting the sum of m exponentials.

Finding the Amplitudes of the Two Exponentials

Once the taus of the sum of exponentials are known, a technique similar to the single-exponential, noise present, case is used to find the amplitudes and the offset. (The corresponding two-exponential case without noise is not shown.) We generate two exponential datasets based on the values of τ_1 and τ_2 just calculated; both datasets have unity amplitude and zero offset:

$$g_{\tau_1}(t_i) = 1e^{-t_i/\tau_1} + 0 \quad \text{and} \quad g_{\tau_2}(t_i) = 1e^{-t_i/\tau_2} + 0$$

Recalling from Equation 12 that the transform of f is the scaled transform of $g_{\tau 1}$ and $g_{\tau 2}$, the resulting coefficients of each of these datasets is scaled and added together. In the presence of noise, however, this relationship is not exactly true. Rather:

$$d_j(f) \cong a_1 d_j^{\tau_1}(g_{\tau_1}) + a_2 d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 0$$

Linear regression of this equation yields the best possible values of a_1 and a_2 that satisfy:

$$\chi^2 = \sum_{j=1}^{n} \left(d_j(f) - a_1 d_j^{\tau_1}(g_{\tau_1}) - a_2 d_j^{\tau_2}(g_{\tau_2}) \right)^2 \qquad (22)$$

Solution of this equation is not shown, but involves expanding the square inside the summation, minimizing first with respect to a_1 and then to a_2, and solving the resulting simultaneous set of equations for a_1 and a_2.

Finding the Offset

Finally, to find the offset, a formula similar to Equation 8 is used; no regression is needed:

$$a_0 = d_0(f) - a_1 d_0^{\tau_1}(g_{\tau_1}) - a_2 d_0^{\tau_2}(g_{\tau_2}) \qquad (23)$$

Fitting the Sum of Three or More Exponentials

Fitting the sum of three or more exponentials is a simple extension of the fitting the sum of two exponentials case. A full description is not given here.

Speed of Fitting

In tests the Chebyshev technique fit speed was completely unaffected by noise. There was a slight dependence on the number of exponentials being fit.

In contrast to these results, with iterative techniques, fitting a sum of two exponentials usually requires twice as much time, fitting a sum of three exponentials requires three times as much time, and so on.

Goodness of Fit

In tests comparing the Chebyshev method to the Simplex iterative search method, both yielded the same values of the fit parameters in low-noise and no-noise conditions. The tests added varying amounts of noise to exponentials generated from known values. As the noise levels increased, these two methods produced slightly different values of the fit parameters for each test case, although the average of the parameters was the same. At extremely high levels of noise (the peak-to-peak noise reached 30% of the peak of the exponential), the Chebyshev search clearly did not fit as well as the Simplex method, in those times that the iterative Simplex converged at all.

Like all fitting methods, the Chebyshev method fits sums of exponentials data best if the dataset spans several times the largest time constant in the exponential. Although the Chebyshev method consistently outperforms other iterative techniques in this regard, even it can generate incorrect fits in this situation (e.g. trying to fit an exponential function with a time constant of 2000 ms to a dataset spanning just 10 ms!) In particular, as the amount of noise increases, its ability to fit an insufficiently sampled dataset decreases.

The Chebyshev method performs most poorly when fitting data with extremely low-frequency signals present. This may occur under the following circumstances: 60 Hz noise present (or other low-frequency noise) or insufficiently averaged data (as may occur by forming a macroscopic current from an insufficient number of single-channel records: i.e. the single channel events may still be seen). This occurs since low-frequency noise will appear most strongly in the low-index Chebyshev coefficients, the same coefficients that contain most of the information of the exponential. Although iterative techniques do not perform well in this case either, when they converge they do so to the correct result more often than the Chebyshev technique.

Success of Fitting

The Chebyshev technique very rarely fails to find a fit to the data when fitting exponentials or sums of exponentials, as it uses mathematical relationships and linear regression techniques. Experimentally though, it sometimes fails to find a good fit (see above conditions) and can even fail altogether with some kinds of data (for example, if the *x* and *y* data values are identical). In particular, since the Chebyshev technique always finds an answer so quickly, it is tempting to assume that this answer is always correct. Be sure to always compare the fitted data to the dataset.

Note that the Chebyshev method can fail when fitting shifted Boltzmann or exponential power functions, although the failure in these cases is expected to be very rare. In the event of a failure, all function parameters will reported as zero.

Fitting Sums of Exponentials to Non-Evenly Spaced Datasets

At present, the Chebyshev method only works with datasets having equally spaced points, since the relations derived here depend upon that assumption. With datasets that were sampled at two different rates (split-clock acquisition), you must first use the Analyze > Interpolation command to fill in the missing data points using linear interpolation.

Note

At the time of this writing the Chebyshev search method had not been published in a peer-reviewed journal. While we have conducted many empirical tests that have confirmed the accuracy of the fits, you should occasionally compare Chebyshev-fitted results to those of one or more of the other fitting methods.

MAXIMUM LIKELIHOOD ESTIMATION

Data are fitted by maximizing the logarithm of the likelihood with respect to a set of fitting parameters. Exponentially distributed data (t_i) are described by one or more components (k) of the form:

$$F(t_i) = \sum_{j=1}^{k} a_j e^{-t_i/\tau_j} \qquad (1)$$

where $t_1, t_2, \ldots t_n$ are the n measured data points and τ_j is the time constant of the j^{th} component. Each a_j is a fraction of the total number of events represented by the j^{th} component where:

$$\sum a_j = 1.0$$

The probability density function $f(t_i)$ is obtained by taking the first derivative of Equation 1 with respect to t_i, where, for each t_i:

$$f(t_i) = \frac{d}{dt_i}F(t_i) = \sum_{j=1}^{k} a_j \tau_j e^{-t_i/\tau_j} \qquad (2)$$

The likelihood (L) of obtaining a given set of observed data t_i, given the form of the distribution and the set of function parameters (denoted here by θ) is the product of the probabilities of making each of the N observations:

$$L = \prod_{j=1}^{N} f(t_i|\theta)$$

As the likelihood typically takes on very small values the numerical evaluation of its logarithm is preferable. In practice, limited frequency resolution of the recording system makes it impossible to record events shorter than t_{min}. To generalize this case it is also assumed that no events longer than t_{max} can be measured. Taking these corrections into consideration, the conditional PDF is given by:

$$L(\theta) = \sum_{i=1}^{N} ln[f(t_i|\theta)/p(t_{min},t_{max}|\theta)]$$

where:

$$p(t_{min},t_{max}|\theta) = Prob(t_{min} \le t < t_{max})$$

is the probability that the experimentally measurable dwell-times fall within the range delimited by t_{min} and t_{max} given the probability distribution with parameters θ. In Clampfit t_{min} and t_{max} are defined by the lower and upper limits of the data’s time base.

The fitting algorithm used here is not, strictly speaking, a maximum likelihood estimation. It is an iteratively re-weighted least squares fit to the number of elements in a bin (which should converge on the maximum likelihood estimates of the parameters in this case). The weighting here has to do with the expected variance of the number of elements in a bin, which is Poisson distributed. So the weighting factor is the inverse of the variance (remember that mean = variance in a Poisson distribution). The iterative aspect has to do with the fact that the number of elements per bin gets moved around to correct for censoring (i.e. short events are not measurable). Because of the simple form of the “weighted” sum of exponentials (that’s a different “weight”), the derivatives of the minimized function with respect to each parameter can be written directly and corrected for this Poisson-distributed variance.

Maximum Likelihood for Binned Data

The log likelihood L_b of observing a particular set of bin occupancies n_i for a given set of dwell-time data of times t_i is calculated by:

$$L_b(\theta) = \sum_{i=1}^{N} n_i \; ln\left\{\frac{F(t_{i+1}|\theta) - F(t_i|\theta)}{p(t_s,t_k|\theta)}\right\} \qquad (1)$$

where $F(t_i)$ and $F(t_{i+1})$ are the probability distributions at the lower and upper bounds of the i^{th} bin using the parameter values θ, and $p(t_s, t_k)$ is the probability that the experimental dwell-times fall within the range (k) of the histogram where

$$p(t_s, t_k|\theta) = F(t_s|\theta) - F(t_k|\theta) \qquad (2)$$

In Clampfit, the probability distribution function (Equation 1), rather than the probability density function (Equation 2), is used in the calculations. $F(t)$ is evaluated at each bin edge. The difference gives the probability that an event falls in the bin. This is equivalent to integrating the PDF over the width of the bin.

For a sum of m exponential components the probability distribution function is given by:

$$F(t|\theta) = 1 - \sum_{i=1}^{m} a_i e^{-t/\tau_i}$$

where θ is the entire set of coefficients comprising the fraction of the total number of events in each i^{th} component a_i and the time constants τ_i. The coefficients a_i sum to unity, where:

$$\sum_{i=1}^{m} a_i = 1$$

Therefore, the set of parameters θ has $2m$–1 degrees of freedom. Clampfit can use either the Simplex or variable metric search methods to find the set of parameters that maximize $L_b(\theta)$. Note, however, that only the variable metric method constrains the coefficients a_i to sum to unity. In the Simplex method, these parameters are not constrained and the set of parameters has $2m$ degrees of freedom.

Maximum likelihood will operate on either logarithmically binned or conventionally binned histograms. It is, however, recommended that logarithmically binned data be used if MLE is the fitting method. With conventional binning, it is often not possible to select a bin width that can represent the data satisfactorily if the time constants are widely spaced. This problem can be avoided by binning the data into the variable-width bins of a logarithmic histogram. For detailed discussion see Sigworth & Sine (1987) and Dempster (1993).

The EM Algorithm

The EM (Expectation step – Maximization step) algorithm computes maximum likelihood parameter estimates for binned observations from a mixture of exponentials. This algorithm, described in Dempster *et al.* (1977), is used in Clampfit to estimate initial seed values for maximum likelihood estimates for fitting exponential probability functions.

MODEL COMPARISON

In general, with non-linear fitting routines, when the order of the fitting function is increased the fitted curve will appear to improve (up to a point, of course, until it becomes difficult to distinguish between successive order fits). Although it is often possible to choose one fit over another by visual inspection, this is not always the case. Moreover, visual inspection is not an adequate substitute for a statistical comparison, especially if two different models (i.e. different orders) produce fits that are visually very similar. Clampfit provides a means of statistically comparing different models that have been fitted with either a least-squares routine or with maximum likelihood (Horn, 1987, Rao, 1973).

When model comparison is selected, the text "compare models" appears next to the function name above the equation window. If the selected function does not support model comparison, the model comparison status text is blanked out.

Note that fixed fitting function parameters are not allowed when comparing models.

Maximum Likelihood Comparison

Suppose you wish to compare two models, F and G, which have been fitted using the maximum likelihood method. The probability densities for these models are given by $f\,(x,\ \theta)$ and $g\ (x,\ \beta)$ where x is the dataset, and β and θ are the function parameters with dimensions kf and kg (where $kg > kf$). The natural logarithm of the likelihood ratio (LLR) for F and G is defined as:

$$LLR = log\left\{\frac{sup_{\beta}(x,\ \beta)}{sup_{\theta}(x,\ \theta)}\right\} = log\left\{\frac{g(x,\ \beta)}{f(x,\ \theta)}\right\}$$

where β and θ are the parameter values (maximum likelihood estimates) that maximize the likelihood for each probability density. The suprema of $f\ (x,\ \theta)$ and $g\ (x,\ \beta)$ are denoted by $sup_{\theta}\ (x, \theta)$ and $sup_{\beta}\ (x, \beta)$, respectively. When model F is true, 2LLR has a Chi-square distribution with $k_g - k_f$ degrees of freedom (Rao, 1973; Akaike, 1974, Horn, 1987).

For example at a confidence level of 0.95 (p < 0.05) for $k_g - k_f = 2$ degrees of freedom (as is always the case between successive models) Chi-square is 5.991. Therefore, if 2LLR < 5.991 then it is assumed that model G does not represent a significant improvement over model F and model F is selected as the best fit.

Least Squares Comparison

In the case of least squares fitting (Simplex or Levenberg-Marquardt fitting methods) the SSE for models F and G is defined as:

$$SSE_f = \sum_{i=1}^{n} [x_i - f(x_i|\theta)]^2 \qquad \text{for model } F$$

and:

$$SSE_g = \sum_{i=1}^{n} [x_i - g(x_i|\beta)]^2 \qquad \text{for model } G$$

where x_i are the n data points, $f(x_i\ |\ \theta)$ and $g(x_i\ |\ \beta)$ are the values predicted by models F and G, respectively, and θ and β are the set of function parameters that minimize SSE.

To compare models *F* and *G* a form of the likelihood ratio test is used. In the case of least squares fitting the statistic *T* (Rao, 1973) is defined by

$$T = \frac{(SSE_f - SSE_g)}{SSE_g} \cdot \frac{(n - k_g)}{k_f}$$

where SSE_f and SSE_g are the sums of squared errors for models *F* and *G*, respectively. *T* has an *F*-distribution which can be compared with a standard *F*-distribution table, with k_f and $n–k_g$ degrees of freedom. This statistic is denoted in Clampfit by “F”.

Note that the degrees of freedom (k_f and $n–k_g$) will be different for each successive model.

DEFINING A CUSTOM FUNCTION

The rules for defining a custom function and associated issues are as follows:

- Only one dependent variable is allowed. The dependent variable is already expressed, as is the equal sign, so these should not be entered as part of the equation. Only the right-hand side of the equation should be specified.
- When fitting from Analysis window data, only one independent variable is allowed. This variable must be expressed as x1 (or X1).
- When fitting from a Results window, up to six independent variables are allowed. These variables must be expressed as x1...x6 (or X1...X6).
- The maximum number of function parameters is 24. Parameter names must be expressed as p1...p24 (or P1...P24).
- The maximum length of the function including parenthesis is 256 characters.
- Parameter labels (*p*), independent variable labels (*x*) and mathematical operations such as log, sin, etc. are case-insensitive, so you may use either lower or upper case when specifying these labels or operations.
- Automatic seeding is not available for custom functions. You must specify all seed values before the fit will commence. If you try to commence fitting (by clicking OK) before all parameter seeds have been specified you will receive an error message.
- Graphically assisted seeding is not currently available when fitting to data from a Results window. If you absolutely require graphical seeding for fitting Results window data, you can first save the data to an ATF file, then import the ATF data into an Analysis window for graphical seeding and fitting. However, the data can contain only positive, uniformly spaced independent variable values (X axis data) for proper display in the Analysis window. Also, keep in mind that only a single independent variable can be specified for fitting from the Analysis window.

The custom function is compiled when switching from the Function>Methods tab to either of the other tabs in the fitting dialog. Once compiled successfully, the equation will appear on the line above the equation window in the fitting dialog. If there is an error in the expression, compiler warnings will be issued.

Be careful with parentheses. For example, be aware that 2*x1+p2 is not the same as 2*(x1+p2). In the former case 2*x1 is evaluated before p2 is added to the product, whereas in the latter case, x1+p2 is evaluated before the multiplication is performed.

MULTIPLE-TERM FITTING MODELS

A fitting model can be expressed as a sum of identical functions (terms). Each term in a multiple-term model will be assigned a "weight" or "amplitude" that will reflect the contribution of that term to the fitted curve (Schwartz, 1978). For example, a two-term standard exponential will be of the form

$$f(x) = A_1 e^{-t/\tau_1} + A_2 e^{-t/\tau_2} + C$$

where A_1 and f_1 are the amplitude and time constant, respectively, for the first term and A_2 and f_2 are the amplitude and time constant, respectively, for the second term. The variable C is a constant offset term along the Y axis.

Multiple terms for custom functions must be explicitly defined within the custom function itself, for example, the above two-term exponential function would be specified as

$$f(x) = p1 \cdot exp(-x1/p2) + p3 \cdot exp(-x1/p4) + p5$$

where $p1$ and $p3$ are the amplitudes, $p2$ and $p4$ are the time constants and $p5$ is the constant offset term.

MINIMIZATION FUNCTIONS

Sum of Squared Errors

Levenberg-Marquardt, variable metric and Simplex only.

The function to be minimized is

$$SSE = \sum_{i=1}^{N} (y_i - y)^2 = \sum_{i=1}^{N} [y_i - f(x, P)]^2$$

where SSE (sum of squared errors) is the sum of the squares of the difference between the data y_i and the fitting function $y = f(x,P)$ with a set of parameters P to be fitted over N data points. The optimal values for P are assumed to occur when SSE is at a minimum. Weighting may or may not be applied to modify this function.

Maximum Likelihood Minimization

Variable metric and Simplex only.

Maximum likelihood estimation (MLE) is available only for the standard and log-transformed probability exponential functions and only with the variable metric or Simplex search methods. Moreover, these functions are intended to be used with binned data. That is, the dependent variable values (X axis data) are assumed to be bin center values.

Strictly speaking, the likelihood is maximized so the use of "minimization method" might be deemed inappropriate here. However, the fitting method minimizes the negative of the log likelihood value, which is equivalent to maximizing the positive log likelihood.

See *"Maximum Likelihood Estimation"* on page 201 for a description of the algorithm.

Mean Absolute Minimization

Levenberg-Marquardt and Simplex only.

Mean absolute minimization is a linear criterion that weights a badly-fitted data point proportionately to its distance from the fitted curve, rather than the square of that distance. If some points have substantially more error than others then the best sum of squares fit might deviate from the more reliable points in an attempt to fit the more reliable ones. The mean absolute error fit is influenced more by the majority behavior than by remote individual points. If the data are contaminated by brief, large perturbations, mean absolute error minimization might perform better than sum of squares minimization.

The function E to be minimized is

$$E = abs\left(\sum_{i=1}^{N}(y_i - y)\right) = abs\left(\sum_{i=1}^{N}[y_i - f(x, P)]\right)$$

Minimax Minimization

Levenberg-Marquardt and Simplex only.

Minimax minimization yields a fit in which the absolute value of the worst-fitted data point residual is minimized. This might be useful for certain datasets when the fit must match the data within a certain tolerance.

The function E to be minimized is

$$E = abs(max(y_i - y)) = abs(max[y_i - f(x, P)])$$

where $max\ (y_i - y)$ is the largest absolute difference between the data and the fit.

WEIGHTING

For the search methods that support least squares minimization (Levenberg-Marquardt, variable metric and Simplex)) the sum of the squares (SSE) of the difference between the data (f_i^{obs}) and fitted curve ($f_i(q)$) is minimized, where

$$SSE = \sum(f_i^{obs} - f_i(\theta))^2$$

This works quite well when uncertainties are random and not related to the time or number of counts. However, the uncertainty observed is often a function of the measured value *y*, such that for larger values of y there are substantially greater deviations from the fit than for smaller values of *y*. To compensate for this, weighting becomes quite useful during the assessment of the fit, where:

$$SSE = \sum\left\lfloor\left(\left(f_i^{obs} - f_i(\theta)\right)^2 / f(\theta)\right)\right\rfloor$$

The Levenberg-Marquardt method is the only fitting method that supports weighting. The SSE minimization function can be weighted in one of the following four ways:

None

In this case the denominator *f* (θ) is 1.

Function

Weighting the sum of squared errors function generates the Chi-square (χ^2) function. The χ^2 value for a function with a given set of parameters θ is given by:

$$\chi^2 = \sum_{i=1}^{m} \frac{f_i^{obs} - f_i(\theta)}{f_i(\theta)}$$

where *m* is the number of points in the histogram fitting range, f_i (θ) is the fit function calculated for the i^{th} data point, and f_i^{obs} is the observed value of the i^{th} point.

Data

This is a modified form of the χ^2 function, where:

$$\text{Modified } \chi^2 = \sum_{i=1}^{m} \frac{f_i^{obs} - f_i(\theta)}{f_i^{obs}}$$

where *m* is the number of points in the histogram fitting range, f_i (θ) is the fit function calculated for the i^{th} data point, and f_i^{obs} is the observed value of the i^{th} point.

Bin Width

Weighting by the bin width weights the sum of squared errors by the width of each i^{th} histogram bin, such that:

$$SSE_b = \sum_{i=1}^{m} \frac{f_i^{obs} - f_i(\theta)}{(x_i - x_s)}$$

where x_i is the right bin edge value and x_s is the left bin edge value. Bin width weighting is allowed only for the predefined log-transformed exponential function that assumes log-binned data. It is not available for custom functions.

The selected weighting type is also applied to mean absolute and minimax minimization.

Note that if you wish to apply different weighting criteria you can export the data to a results sheet, modify the data using column arithmetic and subsequently fit the data directly from the results sheet.

NORMALIZED PROPORTIONS

The normalized proportion is the absolute value of that parameter divided by the sum total of the absolute values of all of the proportion terms, that is:

$$P_{norm_k} = \frac{P_{abs_k}}{\sum_{j=1}^{n} P_{abs_j}}$$

Normalized proportions are most likely to be specified with the exponential probability functions or the Exponential, weighted/constrained function. The variable metric method (and only this method) constrains the proportion terms to sum to 1.0 during fitting (see *"The Variable Metric Method"* on page 186), but only for the standard or log-transformed exponential probability functions when using maximum likelihood or the weighted/constrained exponential with least squares.

ZERO-SHIFTING

In cases where the natural origin for a dataset is ambiguous or inappropriate, it would be desirable to shift the origin of the data that are to be fitted to zero. For example, if a time constant of an exponential curve is to be extracted from an arbitrary segment of a large dataset it would be reasonable to force the selected fitting range to start at zero. In the same vein, it might also be desirable to set a zero origin just after a stimulus artifact. To this end, the "zero-shift" option is provided. If zero-shift is enabled then for a set of i data points x_i, each point x is offset such that $x = x_i - x_0$ where x_0 is the value of the first data point.

However, it is important to note that zero-shifting can affect the values of some fitted parameters. Consequently, in some instances, zero-shifting the data might not be appropriate and could lead to unexpected results. For example, when fitting a "Z-delta Boltzmann" to current/voltage data the parameter V_{mid} (the voltage at which the current is half-maximal) will differ depending on whether or not zero-shift is enabled. In the following example, the data were zero-shifted prior to fitting.

Figure 11.3 shows a fit of the Z-delta Boltzmann function to current/voltage data. The fitted curve is the line without symbols.

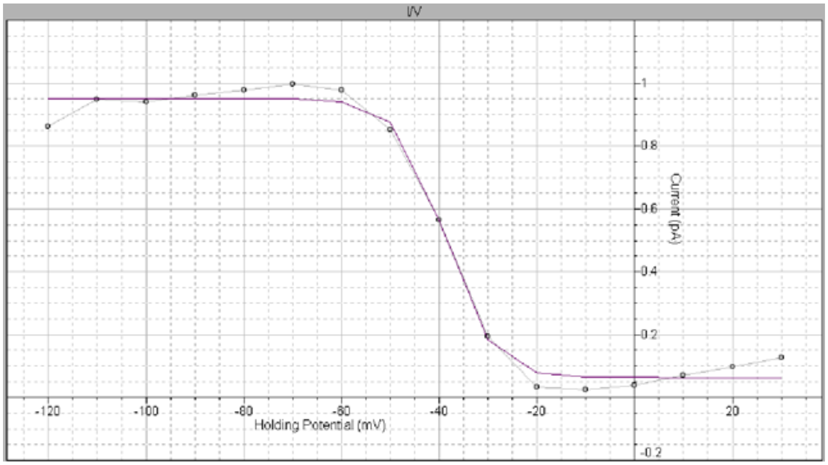


Figure 11.3: Fit of Z-delta Boltzmann function to current/voltage data.

In this fit V_{mid} is reported as +81.16 but from the data one would expect V_{mid} to be about –40 mV. Nevertheless, the value of +81.16 is in fact correct because the first point (originally at –120 mV) has been forced to zero such that the actual fitting range is from 0 to +140 mV. On this scale, the reported positive value is a reasonable estimate of V_{mid}.

If zero-shift is disabled, the fitted curve looks exactly the same but V_{mid} is now reported as –38.84 mV. The other parameters of this function are identical in both cases. As the expected value for this dataset is indeed in the range of –40 mV, it is clear that in this particular case it would not be appropriate to use zero-shifting. In fact, if zero-shift is inadvertently enabled it would appear that the fitting routine is malfunctioning, which is not the case. It is, therefore, very important that the status of the zero-shift option be known at all times when fitting.

12. Fitting Functions

Clampfit provides a large selection of predefined fitting functions to assist in data analysis. Some functions require that restrictions, such as forced positive parameters, are applied during the fit. Accordingly, some options in the Fit dialog might be automatically set to comply with mandatory fitting requirements. Furthermore, some functions require automatic preprocessing of the data, such as normalization, prior to fitting. In the function descriptions below, any automatic data preprocessing and forced option requirements for a function are listed on the line above the function formula.

The references cited provide sources for more detailed explanation of the functions and examples of specific experimental applications.

BETA FUNCTION

$$f(x) = x^{a-1}(1-x)^{b-1}/B(a,b) + C$$

$$B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx$$

$$a = \alpha\tau, \qquad b = \beta\tau$$

→ Requires a normalized X axis.

→ All parameters except the offset (*C*) are forced positive.

The beta function is used to describe the steady state filtered amplitude distribution of a two-state process, where α and β are rate constants and τ is the time constant of a first-order filter. This method has been used to measure block and unblock rates in the microsecond time range even under conditions when individual blocking events were not time-resolved by the recording system (Yellen, 1984).

This function describes a probability distribution that requires the X axis data range to be 0 to 1. If the data do not meet this criterion the X axis values are automatically normalized prior to fitting. These values are rescaled after fitting so that the fitted curve will conform to the original data.

The beta function is intended to be used primarily with data that have been imported into a Results or Graph window, and therefore has limited utility with respect to time-based Analysis window data.

The fit solves for parameters a, b, $B(a,b)$ and the constant y-offset C. The rate constants α and β can be obtained by dividing a and b, respectively, by the filter time constant, τ. Clampfit does not perform this calculation.

The recommended fitting method is Levenberg-Marquardt.

BINOMIAL

$$f(x) = \frac{n!}{(n-x)!x!}P^x(1-P)^{n-x}$$

→ Requires an integral X axis.

→ Requires normalized data.

→ Maximum number of points = 170.

→ The parameter P is forced positive.

The binomial distribution describes the probability, P, of an event occurring in n independent trials. For example, this function can be used to determine the probability for a particular number of n independent ion channels being open simultaneously in a patch (Colquhoun, *et al*. 1995). This function has also been applied to quantal analysis of transmitter release (Bekkers, *et al* 1995, Larkman, *et al* 1997, Quastel, 1997).

This function requires integer values of *x*. If the X axis values are not integers then they are converted to such prior to, and rescaled following, fitting. The ordinate data are also normalized so that they range from 0 to 1, and are rescaled after fitting so that the fitted curve conforms to the original data. Rescaling is such that the area under the fitted curve is equal to the area under the original data curve.

The number of sample points for this function is limited to 170 to conform to the computational limit for a factorial.

The binomial function has limited utility with respect to time-based Analysis window data. It is intended to be used primarily with data that have been imported into a Results or Graph window.

The fit solves for the probability variable, *P*. Since the X axis scale is integral the fitted curve will appear as a histogram.

The recommended fitting method is Levenberg-Marquardt.

BOLTZMANN, CHARGE-VOLTAGE

$$f(V) = \frac{I_{max}}{1+e^{(V_{mid}-V)/V_c}} + C$$

This function can be used to examine activation and inactivation profiles of voltage-gated currents (Zhang, *et al.* 1995).

The charge-voltage Boltzmann distribution is given by

$$Q_{on}(E) = Q_{on-max} / [1 + exp((E_{mid} - E)/K)] + C$$

where $Q_{on–max}$ is the maximum charge displaced, E_{mid} is the potential at which $Q_{on} = 0.5 \times Q_{on-max}$, K is the number of millivolts required to change Q_{on} e-fold, and E is the measured potential. C is a constant offset term. This function can be used to fit current-voltage curves of the form:

$$I = I_{max}/[1 + exp((V_{mid} - V)/V_c)]$$

or

$$g = g_{max}/[1 + exp((V_{mid} - V)/V_c)]$$

where V is the membrane potential, V_{mid} is the membrane potential at which the current is half-maximal, and V_c is the voltage required to change I or g e-fold. If I or g are normalized then the data points should be input as I/I_{max} or g/g_{max} and the dependent variable I_{max} or g_{max} in the function above should be fixed to 1.0.

The fit solves for I_{max} (or G_{max}), V_{mid}, V_c and the constant y-offset C.

The recommended fitting method is Levenberg-Marquardt. The variable metric method also works well with this function but is slower to converge.

BOLTZMANN, SHIFTED

$$f(x) = \sum_{i=1}^{n} \frac{A_i}{1 + Be^{-x/\tau_i}} + C$$

Like the standard Boltzmann, this function also defines a sigmoidal curve. A is the amplitude, τ is the "slope" and C is a constant offset in the y direction (see *"Boltzmann, Standard"*, below). However, unlike the standard Boltzmann function, the shifted Boltzmann includes an offset parameter B that shifts the curve along the X axis such that the half-maximal amplitude is at $x = -ln(1/B) \cdot \tau$. Thus, this function is to be used when fitting a sigmoidal curve to data where the half-amplitude point is expected to be displaced from zero along the X axis.

The fit solves for A, B, τ and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt or Chebyshev if fitting only a single term.

BOLTZMANN, STANDARD

$$f(x) = \sum_{i=1}^{n} \frac{A_I}{1 + e^{-x/\tau_i}} + C$$

This function defines a sigmoidal curve. Unlike the shifted Boltzmann function, this function does not include an offset parameter along the X axis.

The physical correlates for A and x are not specified in this equation. It is up to the user to define these quantities. For example, A might be conductance and x might be cell membrane voltage (Bähring, *et al.* 1997). The parameter τ is the slope of the function that specifies the change in x required to produce an e-fold change in A. Note that A is half-maximal at $x = 0$ (where $f(x) = A/(1 + e^{-0/\tau}) = A/2$). Consequently, the fitted curve is sigmoidal only if there are both positive and negative x data with the half-amplitude at or very close to zero. If the half-amplitude is offset significantly from zero the shifted Boltzmann function should be used.

The fit solves for the amplitude A, the width τ and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt.

BOLTZMANN, Z-DELTA

$$f(V) = V_{min} + \frac{V_{max} - V_{min}}{1 + e^{\frac{Z_d F}{RT}\langle V - V_{mid}\rangle}}$$

This function can be used to analyze the voltage dependence of gating charges in ion channels (Hille, 1992).

V_{min} and V_{max} are the minimum and maximum voltages, Z_d is the magnitude of the charge valence associated with the electric field V, V_{mid} is the voltage at which $f(V)$ is half-maximal, F is the Faraday constant, R is the Gas constant, T is absolute temperature. The temperature is optionally specified (in °C).

The fit solves for V_{max}, V_{min}, V_{mid} and the constant y-offset C.

The recommended fitting method is Levenberg-Marquardt.

CURRENT-TIME COURSE (HODGKIN-HUXLEY)

$$f(t) = I'(1 - e^{-t/\tau_j})^a (k_\infty - (k_\infty - 1)e^{-t/\tau_k})^b + C$$

This is the Hodgkin-Huxley model for describing the time course of voltage-dependent ionic membrane currents. This equation was originally used to describe voltage-activated sodium and potassium currents (with $a = 3$ and $b = 1$). The term $k_{\bullet}$ is the steady-state inactivation, I' is the maximum current that is achieved in the absence of inactivation, τ_j is the activation time constant, τ_k is the inactivation time constant, and the power terms a and b are empirically determined (Dempster, 1993, pages 140–142).

The fit solves for I', τ_j, τ_k, $k_{\bullet}$ and the constant y-offset C. The power terms a and b must be optionally specified.

The recommended fitting method is Levenberg-Marquardt.

EXPONENTIAL, ALPHA

$$f(t) = \sum_{i=1}^{n} A_i t e^{-t/\tau_i} + C$$

The alpha exponential function has been used to describe temporal responses at the neuronal soma to synaptic input (Gerstner, *et al.* 1992 and Gerstner, *et al.* 1993).

The fit solves for the amplitude A, the time constant τ and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt.

EXPONENTIAL, CUMULATIVE PROBABILITY

$$f(t) = \sum_{i=1}^{n} P_i(1 - e^{-t/\tau_i}) + C$$

This function fits data that have been binned cumulatively. That is, each successive bin contains its own data plus the data in all of the preceding bins.

This function should not be used for binned data because cumulative binning creates artificial correlations between successive bins. The correlation occurs because each successive bin contains all of the data in the preceding bins. The cumulative exponential function provides meaningful results only if the data values are not correlated.

The fit solves for the proportion (amplitude) P, the time constant τ and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt.

EXPONENTIAL, LOG PROBABILITY

$$f(t) = \sum_{i=1}^{n} P_i e^{[ln(t) - ln(\tau_i)]e^{ln(t) - ln(\tau_i)}}$$

→ Can only be used with Results or Graph window data.

→ The dwell-time data (t) must be input as $log_{10}(t)$.

This function describes dwell-time data, usually from single channel experiments, that have been binned on a logarithmic time scale. Logarithmic binning is often preferable to conventional linear binning because of its superior resolution of widely spaced time constants (Sigworth & Sine, 1987). Histograms can be imported from pSTAT or the QUB module MIL.

The fit solves for the proportion (amplitude) P, the time constant τ and the constant y-offset C for each component i.

The recommended fitting method is variable metric with maximum likelihood estimation.

EXPONENTIAL, POWER

$$f(t) = \sum_{i=1}^{n} A_i(1 - e^{-t/\tau_i})^a + C$$

The fit solves for the amplitude A, the time constant τ and the constant y-offset C for each component i. The power term a must be optionally specified.

The recommended fitting method is Levenberg-Marquardt or Chebyshev if fitting only a single term (Chebyshev can solve for a single term only).

EXPONENTIAL, PROBABILITY

$$f(t) = \sum_{i=1}^{n} P_i \tau_i^{-1} e^{-t/\tau_i} + C$$

This function can be used to fit single channel dwell time distributions that have not been converted to Log duration. For each component of the distribution, the fit solves for the proportion P, the time constant τ and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt. Maximum likelihood estimation can also be used with either the variable metric or Simplex fitting methods, but convergence will be slower.

EXPONENTIAL, PRODUCT

$$f(t) = \sum_{i=1}^{n} A_i\left(1 - e^{-t/\tau_{r_i}}\right)\left(e^{-t/\tau_{d_i}}\right) + C$$

This function can be used to fit postsynaptic events (excitatory or inhibitory postsynaptic potentials). The fit solves for the amplitude A, the rise time constant τ_r and the decay time constant τ_d for each component i.

The recommended fitting method is Levenberg-Marquardt.

EXPONENTIAL, SLOPING BASELINE

$$f(t) = \sum_{i=1}^{n} A_i e^{-t/\tau_i} + mc + C$$

This function is used to fit an exponential of the standard form to data that are superimposed on a sloping baseline, for example resulting from a constant baseline drift.

The fit solves for the amplitude A, the time constant τ for each component i, and the common parameters, the slope m and constant y-offset C for each component i.

The recommended fitting method is Chebyshev.

EXPONENTIAL, STANDARD

$$f(t) = \sum_{i=1}^{n} A_i e^{-t/\tau_i} + C$$

This is the most basic function used to fit changes in current or voltage that are controlled by one or more first-order processes. The fit solves for the amplitude A, the time constant τ, and the constant y-offset C for each component i.

The recommended fitting method is Chebyshev.

EXPONENTIAL, WEIGHTED

$$f(t) = K_0\left(\sum_{i=1}^{n} f_i e^{-K_i t}\right) + C$$

This function is identical to the constrained exponential function except that the sum of the f_i components is not constrained to 1.

The fit solves for the proportion (amplitude) f, the rate constant K, the "weight" K_0 and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt.

EXPONENTIAL, WEIGHTED/CONSTRAINED

$$f(t) = K_0\left(\sum_{i=1}^{n} f_i e^{-K_i t}\right) + C \quad \text{where} \quad \sum_{i=1}^{n} f_i = 1.0$$

→ Requires the variable metric fitting method.

This function has been used to describe the recovery rate of ground-state absorption following photo-excitation of intercalated metal complexes bound to DNA (Arkin, *et al.* 1996).

The fit solves for the proportion (amplitude) f, the rate constant K, the weight K_0 and the constant y-offset C for each component i. The f_i terms sum to 1.0.

The fitting method must be variable metric.

GAUSSIAN

$$f(t) = \sum_{i=1}^{n} A_i \frac{e^{-(x-\mu_i)^2/(2\sigma_i^2)}}{\sigma_i\sqrt{2\pi}} + C$$

This is for data that can be described by one or more normal distributions. For *n* components, the fit solves for the amplitude *A*, the Gaussian mean μ, the Gaussian standard deviation σ and the constant *y*-offset *C* for each component *i*.

This function is generally used for describing amplitude distributions of single channel events (Heinemann, 1995).

The recommended fitting method is Levenberg-Marquardt.

GOLDMAN-HODGKIN-KATZ

$$f(x,y,z) = \frac{RT}{F} ln \frac{[X]_1 + \alpha[Y]_1 + \beta[Z]_1}{[X]_2 + \alpha[Y]_2 + \beta[Z]_2} \qquad \alpha = (pY)/(pX) \qquad \beta = (pZ)/(pX)$$

→ This function can only be used with Results window data.

This function is used to describe the steady-state dependence of membrane voltage on ion concentrations and the relative permeability of those ions through the membrane.

The equation assumes that all the ions are monovalent. For positive ions $[X]_1$ refers to the concentration outside the membrane and $[X]_2$ refers to the intracellular concentration. For negative ions $[X]_2$ refers to the concentration outside the membrane and $[X]_1$ refers to the intracellular concentration.

The fit solves for the permeability ratios α and β.

The recommended fitting method is Levenberg-Marquardt.

GOLDMAN-HODGKIN-KATZ, EXTENDED

$$f(x,y,z^{2+}) = \frac{RT}{F} ln \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$a = [X]_2 + 4\beta[Z]_2 + \alpha[Y]_2$$
$$b = [X]_2 - [X]_1 + \alpha([Y]_2 - [Y]_1)$$
$$c = -[X]_1 - \alpha[Y]_1 - 4\beta[Z]_1$$

$$\alpha = pY/pX \qquad \beta = pZ/pX$$

→ This function can only be used with Results window data.

This function is used to describe the steady-state dependence of membrane voltage on ion concentrations and the relative permeability of those ions through the membrane.

However, this formulation extends the Goldman-Hodgkin-Katz relationship to include the effect of a divalent ion such as calcium or magnesium. By measuring the dependence of resting potential or reversal potential on varying concentrations of the relevant monovalent and divalent ions, this equation can be used to calculate the relative permeability of the activated conductance(s) to calcium compared to sodium, potassium, or other experimental ions (Piek, 1975; and Sands, *et al.* 1991).

The fit solves for the permeability ratios α and β.

The recommended fitting method is Levenberg-Marquardt.

HILL (4-PARAMETER LOGISTIC)

$$f(x) = I_{min} + \frac{I_{max} - I_{min}}{1 + (C_{50}/[x])^h}$$

This is a modified form of the Hill equation that is useful for fitting dose-response curves. The half-maximal concentration is determined directly. I_{min} refers to the baseline response, I_{max} refers to the maximum response obtainable with drug x, C_{50} is the concentration at half-maximal response (inhibitory or excitatory) and h is the Hill slope.

The fit solves for I_{min}, I_{max}, C_{50} and h.

The recommended fitting method is Simplex.

HILL, LANGMUIR

$$f(x) = \sum_{i=1}^{n} \frac{I_{max_i}[x]^{h_i}}{C_{50_i}^{h_i} + [x]^{h_i}} + C$$

The Langmuir-Hill equation allows for fitting a sum of Langmuir components. It is useful for fitting data where non-specific binding of the agonist is linear. I_{max} refers to the maximum response obtainable with drug x, C_{50} is the concentration at half-maximal response (inhibitory or excitatory) and h is the Hill slope. C is a constant y-offset.

The fit solves for I_{max}, C_{50}, h and C.

The recommended fitting method is Simplex.

HILL, STEADY STATE

$$f(S) = \frac{V_{max}[S]^n}{K^n + [S]^n} + C$$

This is a general equation that can be applied to many kinds of concentration-dependent pharmacological or ion channel responses. V_{max} refers to the maximum response obtainable with drug S. By definition a partial agonist will have a V_{max} value that is less than the V_{max} of a full agonist. K is indicative of potency but it is not equal to the concentration at half-maximal velocity except when $n = 1$. The value of n places some limitations on the degree of cooperativity of the ligand-dependent processes. In order to define a concentration-dependent inhibitory process, n can be seeded with a negative value.

The fit solves for V_{max}, K and n.

The recommended fitting method is Simplex.

LINEWEAVER-BURK

$$f(S) = \frac{K_m}{V_{max}[S]} + \frac{1}{V_{max}} \qquad \text{where } S = 1/x$$

This equation, derived by taking the reciprocal of both sides of the Michaelis-Menten equation, describes a straight line with a slope of K_m/V_{max} and a y-intercept of $1/V_{max}$. It is useful for gaining information on enzyme inhibition (Lehninger 1970, page157).

The fit solves for K_m and V_{max}.

The recommended fitting method is Levenberg-Marquardt.

LOGISTIC GROWTH

$$f(X) = \frac{R_{max}}{1 + Ae^{-Bx}} + C$$

This function describes an exponential growth that is subject to saturation that occurs at the limiting value R_{max}. The parameter A is the number of times the initial value must grow to reach R_{max}, and B determines the rate and direction of growth. The function will increase when B is positive and decrease when B is negative.

The fit solves for R_{max} and A, B and the constant y-offset C.

The recommended fitting method is Levenberg-Marquardt.

LORENTZIAN DISTRIBUTION

$$f(x) = \sum_{i=1}^{n} \frac{2A_i\omega}{4\pi(x-\mu)^2 + \omega^2} + C$$

The Lorentzian distribution function is generally used to characterize energy transition spectra that exhibit homogenous broadening of the peak. For example, the natural line shape in spectroscopy can be characterized by this function. The peak of a spectral line is narrow but broadens as a result of uncertainties in the energy level of the excited state. It, nevertheless, retains the Lorentzian line shape.

The fit solves for the area A under the curve, the half-width ω of the peak, the X axis center μ of the peak (generally the center frequency) and the constant y-offset C for each component i.

The recommended fitting method is Levenberg-Marquardt.

LORENTZIAN POWER 1

$$S(f) = \sum_{i=1}^{n} \frac{S(O)_i}{1 + (f/f_{c_i})^2}$$

The power spectra of current fluctuations can be described by the sum of one or more Lorentzian functions of this form.

The time constant is related to the cutoff frequency, f_c, at which the power has declined to $S(0)/2$ by

$$\tau = \frac{1}{2\pi f_c}$$

(Dempster 1993 pages 196–197; Stevens 1981).

The fit solves for $S(0)$ and f_c for each component i.

The recommended fitting method is Levenberg-Marquardt.

LORENTZIAN POWER 2

$$S(f) = \sum_{i=1}^{n} \frac{S(O)_i}{1 + (2\pi f \tau_i)^2}$$

The power spectra of current fluctuations produced by ion channels can be described by the sum of one or more Lorentzian functions of this form where:

$$\tau = \frac{1}{1/\tau_o + 1/\tau_c}$$

If the probability of the channel being open is low relative to the probability of the channel being closed then the channel closed time constant τ_c can be ignored and τ can be equated to the channel open time constant τ_o. At low frequencies the function tends toward $S(0)$. At high frequencies the spectrum decays in proportion to f^2 (Dempster, 1993 and Stevens, 1981).

The fit solves for $S(0)$, τ and the constant y-offset C for each component i.

Note that the parameter τ has units of s if the frequency, f, is in Hz.

The recommended fitting method is Levenberg-Marquardt.

MICHAELIS-MENTEN

$$f(S) = \frac{V_{max}[S]}{[S]+K_m} + C$$

This is a general equation to describe enzyme kinetics where *[S]* refers to the concentration of substrate. In this equation, V_{max} refers to rate of catalysis when the concentration of substrate is saturating. K_m refers to the Michaelis constant ($(k_{-1} + k_2)/k_1$) where k_1 and k_{-1} are forward and backward binding rates of the enzyme-substrate complex, respectively, and k_2 is the first-order rate constant for the formation of product from the bound enzyme-substrate complex.

The fit solves for V_{max}, K_m, and the constant *y*-offset *C*.

NERNST

$$V(x) = \frac{RT}{zF} ln \frac{[x]_1}{[x]_2}$$

This function describes the condition where an equilibrium exists between the energy associated with membrane voltage (*V*) and the energy associated with a concentration gradient of an ion species *x*. Hence, the Nernst potential for a given ion is often also referred to as the equilibrium potential for that ion.

The fit solves for the concentration $[x]_2$ given a series of concentrations $[x]_1$.

PARABOLA, STANDARD

$$f(x) = Ax^2 + Bx + C$$

This is the parabolic function.

The fit solves for the parameters *A*, *B* and the constant *y*-offset *C*.

PARABOLA, VARIANCE-MEAN

$$f(x) = ix - x^2/N$$

This is a form of the parabolic function used to describe synaptic event data where *i* is the unitary synaptic current amplitude and *N* is the number of release sites (Clements & Silver, 2000).

The fit solves for the parameters *i*, and *N*.

POISSON

$$f(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

→ Requires an integral X axis.

→ Requires normalized data.

→ Maximum number of points = 170.

→ The data will be automatically zero-shifted.

The Poisson distribution describes the probability of getting x successes with an expected, or average number of successes denoted by λ. This is similar to the binomial function but is limited to cases where the number of observations is relatively small.

This function requires integer values of x. If the X axis values are not integers then they are converted to such prior to, and rescaled following, fitting. The ordinate data are also normalized so that they range from 0 to 1, and are rescaled after fitting so that the fitted curve conforms to the original data. Rescaling is such that the area under the fitted curve is equal to the area under the original data curve.

The number of sample points for this function is limited to 170 to conform to the computational limit for a factorial.

The fit solves for the λ given a series of observed probability values x. Since the X axis scale is integral the fitted curve will appear as a histogram.

The recommended fitting method is Levenberg-Marquardt.

POLYNOMIAL

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

The fit solves for the polynomial coefficients a_i. The term a_0 always exists in the solution. A first-order (or “one term”) polynomial is, therefore, given by $f(x) = a_0 x^0 + a_1 x^1 = a_0 + a_1 x$, which is a straight-line fit.

The maximum order is 6.

STRAIGHT LINE, ORIGIN AT ZERO

$$f(x) = ix$$

This function is used to fit variance-mean data (V-M analysis) to estimate the unitary current, i. The fit solves for the slope i, forcing the origin to zero.

STRAIGHT LINE, STANDARD

$$f(x) = mx + b$$

The straight-line fit solves for the slope m and the y-intercept b.

VOLTAGE-DEPENDENT RELAXATION

$$f(V) = \frac{1}{a_0 e^{V/\alpha} + b_0 e^{-V/\beta}} + C$$

This function describes the relaxation kinetics for a two-state voltage-dependent process. The forward and reverse rate constants are α and β, respectively. The term a_0 is value of α at $V = 0$ and b_0 is value of β at $V = 0$.

The fit solves for α, β, a_0, b_0 and the constant y-offset C.

The recommended fitting method is Levenberg-Marquardt.

CONSTANTS

F = Faraday constant = $9.648456 \, x \, 10^4$ C/mol.

R = gas (Rydberg) constant = 8.31441 J/mol-deg K

A. References

PRIMARY SOURCES

Abramowitz, M. and Stegun, I.A., eds. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover Publications, New York, 1972.

Akaike, H. "A new look at statistical model identification." *IEEE Transactions on Automatic Control* AC-19, 1974.

Arkin, M.R., Stemp, E.D.A., Holmlin, R.E., Barton, J.K, Hörmann, A., Olson, E.J.C. and Barbara, P.F. "Rates of DNA-mediated electron transfer between metallointercalators." *Science* 273, 475–480, 1996.

Bähring, R., Bowie, D., Benveniste, M. and Mayer, M.L. "Permeation and block of rat GluR6 glutamate receptor channels by internal and external polyamines." *J. Physiol.* 502, 575–589, 1997.

Bekkers, J.M. and Stevens, C.F. "Quantal analysis of EPSCs recorded from small numbers of synapses in hippocampal cultures." *J. Neurophysiol.* 73, 1145–56, 1995.

Clements, J.D. and Bekkers, J.M. "Detection of Spontaneous Synaptic Events with an Optimally Scaled Template." *Biophysical Journal* 73, 220–229, 1997.

Clements, J.D. and Silver, R.A. "Unveiling Synaptic Plasticity: A New Graphical and Analytical Approach." *Trends in Neurosciences* 23, 105–113, 2000.

Colquhoun, D. and Hawkes, A.G. "The Principles of the Stochastic Interpretation of Ion-Channel Mechanisms" in: *Single-Channel Recording*, Second Edition, eds. Sakmann, B. and Neher, E. Plenum Press, New York. 432, 1995.

Colquhoun, D and Sigworth, F.J. "Fitting and Statistical Analysis of Single-Channel Records." Chap. 19 in *Single-Channel Recording*, Second Edition, eds. Sakmann, B. and Neher, E. Plenum Press, New York, 1995.

Dempster, A.P., Laird, N.M. and Rubin, D.B. "Maximum likelihood from incomplete data via the EM algorithm." *J. R. Statist. Soc. B*, 39,1–38, 1977.

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

Donaldson, J.R. and Tryon, P.V. The Standards Time Series and Regression Package National Institute of Standards and Technology (formerly the National Bureau of Standards), Internal Report *NBSIR* 86–3448, 1990.

Frazier, J.L. and Hanson, F.E. “Electrophysiological recording and analysis of insect chemosensory responses” in: *Insect-Plant Interactions*, eds. Miller, J.R. and Miller, T.A. Springer Verlag, New York, pp. 285–330, 1986.

Geddes, L.A. *Electrodes and the Measurement of Bioelectric Events*. John Wiley and Sons, New York, 1972.

Gerstner, W., and van Hemmen, J.L. “Associative Memory in a Network of ‘Spiking’ Neurons.” *Network* 3, 139–164, 1992.

Gerstner, W., Ritz, R., and van Hemmen, J.L. “A Biologically Motivated and Analytically Soluble Model of Collective Oscillations in the Cortex: I. Theory of Weak Locking.” *Biol. Cybern.* 68, 363–374, 1993.

Heinemann, S.H. “Guide to Data Acquisition and Analysis” in: *Single-Channel Recording*, Second Edition, eds. Sakmann, B. and Neher, E. Plenum Press, New York. pp. 68–69, 1995.

Hille, B. *Ion Channels of Excitable Membranes*, Second Edition. Sinauer Associates, Massachusetts 1992.

Horn, R. “Statistical Methods for Model Discrimination: Applications to Gating Kinetics and Permeation of the Acetylcholine Receptor Channel.” *Biophys.J.* 51, 255–263, 1987.

Kaissling, K.E. “Pheromone deactivation catalyzed by receptor molecules: a quantitative kinetic model.” *Chem. Senses* 23:385–395, 1998.

Kaissling, K.E. and Thorson, J.T. “Insect olfactory sensilla: structural, chemical and electrical aspects of the functional organization” in: *Receptors for Neurotransmitters, Hormones and Pheromones in Insects*, eds. Satelle, D.B., Hall L.M., Hildebrand, J.G. Elsevier/North-Holland Biomedical Press, Amsterdam, pp. 261–282, 1980.

Kreyszig, E. *Advanced Engineering Mathematics*, 5th edition. John Wiley and Sons, New York, 1983.

Larkman, A.U., Jack, J.J. and Stratford, K.J. “Quantal analysis of excitatory synapses in rat hippocampal CA1 in vitro during low-frequency depression.” *J. Physiol.* (Lond.) 505, 457–71, 1997.

Legéndy, R.C. and Salcman, M. “Bursts and recurrences of bursts in the spike trains of spontaneously active striate cortex neurons.” *J. Neurophysiol.* 53(4), 927–939, 1985.

Lehninger, A.L. *Biochemistry.* Worth, New York p. 157. (1970).

Lynn, P.A. and Fuerst, W. *Introductory Digital Signal Processing with Computer Applications*, Revised Edition. John Wiley and Sons, New York. 1994.

Nelder, J.A. & Meade, R. “A Simplex Method for Function Minimization.” *Comput J.* 7, 308–313, 1965.

Okada, Y., Teeter, J.H., and Restrepo, D. “Inositol 1,4,5-trisphosphate-gated conductance in isolated rat olfactory neurons.” *J. Neurophys.* 71(2):595–602, 1994.

Penner, R. “A practical guide to patch clamping” in: *Single-Channel Recording*, Second Edition, eds. Sakmann, B. and Neher, E. Plenum Press, New York, pp. 3–52, 1995.

Piek, T. “Ionic and Electrical Properties” in: *Insect Muscle*, ed. Usherwood, P.N.R. Plenum Press, New York. pp. 281–336. 1975.

Powell, M.J.D. “A fast algorithm for non-linearly constrained optimization calculations” in: *Proceedings of the 1977 Dundee Conference on Numerical Analysis*, ed. Watson, G.A. Springer Verlag. 1978.

Press, W.H., Teukolsky, S.A., Vetterling, W.T and Flannery, B.P. *Numerical Recipes in C. The Art of Scientific Computing*, Second Edition. Cambridge University Press, Cambridge. 1992.

Quastel, D.M. “The binomial model in fluctuation analysis of quantal neurotransmitter release.” *Biophys. J.* 72, 728–53, 1997.

Rao, C.R. *Linear Statistical Inference and Its Applications*, Second Edition. John Wiley Publications. Ch. 4 and 5, 1973.

Reid, C.A. and Clements, J.D. “Postsynaptic Expression of Long-Term Potentiation in the Rat Dentate Gyrus Demonstrated by Variance-Mean Analysis.” *Jnl. of Physiology* 518.1, 121–130, 1999.

Sands, S.B. and Barish, M.E. “Calcium permeability of neuronal nicotinic acetylcholine receptor channels in PC12 cells.” *Brain Res.* 560, 38–42, 1991.

Schnuch, M. and Hansen, K. “Sugar sensitivity of a labellar salt receptor of the blowfly Protophormia terraenovae.” *J. Insect Physiol.* 36(6):409–417, 1990.

Schreiner, W. Kramer, S.K. & Langsam, Y. “Non-linear Least-squares Fitting.” *PC Tech Journal* May, 1985.

Schwartz, G. “Estimating the dimensions of a model.” *Ann. Statistics* 6, 461–464, 1978.

Sigworth, F. and Sine, S.M. “Data transformation for improved display and fitting of single-channel dwell time histograms.” *Biophys. J.* 52, 1047–1054, 1987.

Sokal, R.R. and Rohlf F.J. *Biometry*, 2nd Edition. Freeman, San Francisco, 1981.

Stephens, M.A. *Journal of the Royal Statistical Society, ser. B*. 32, 115–122, 1970.

Stevens, C.F. “Inferences about molecular mechanisms through fluctuation analysis” in: *Membranes, Channels, and Noise*, eds. Eisenberg, R.S., Frank, M., and Stevens, C.F. Plenum Press, New York. pp. 1–20., 1981.

Tang, C.Y. and Papazian, D.M. “Transfer of voltage independence from a rat olfactory channel to the Drosophila ether-à-go-go K^+ channel.” *J. Gen. Physiol.* 109:301–311, 1997.

Thurm, U. “The generation of receptor potentials in epithelial receptors” in: *Olfaction and Taste IV*, ed. Schneider, D. Wissenschaftliche Verlagsgesellschaft, Stuttgart, pp. 95–101, 1972.

Vermeulen, A. and Rospars, J.P. “Dendritic integration in olfactory sensory neurons: a steady-state analysis of how the neuron structure and neuron environment influence the coding of odor intensity”. *J. Comput. Neurosci.* 5:243–266, 1998.

Yellen, G. “Ionic permeation and blockade in Ca^{2+}-activated channels of bovine chromaffin cells.” *J. Gen. Physiol.* 84, 157–186, 1984.

Zhang, L. and McBain, C.J. “Voltage-gated potassium currents in stratum oriens-alveus inhibitory neurones of the rat CA1 hippocampus.” *J. Physiol.* (Lond.) 488, 647–660, 1995.

FURTHER READING

Bishop, O.N. *Statistics for Biology*. Longmans, London, 1966.

Finkel, A.S. “Progress in instrumentation technology for recording from single channels and small cells” in: *Cellular and Molecular Neurobiology*, eds. Chad, V. and Wheal, H. Oxford University Press, New York, pp. 3–25, 1991.

Hamill, O.P., Marty, A., Neher, E., Sakmann, B., and Sigworth, F.J. “Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches.” *Pflügers Arch.* 391:85–100, 1981.

Sakmann, B. and Neher, E., eds. *Single-Channel Recording*, Second Edition. Plenum Press, New York, 1995.

B. Troubleshooting

SOFTWARE PROBLEMS

If you have any installation questions or problems, first check our Clampex or Clampfit web site through the Help > Web Tech Notes command within each application. You will find information regarding compatibility and configuration issues, as well as information of potential bugs and bug fixes. We are continually improving our products, and often a problem that you are having has already been fixed in a newer version of the program. If you think that you have a problem due to a software bug, you can download the latest version of Clampex or Clampfit from the web site via the Help > Web Updates command. Or, for access to a wider range of issues, the Molecular Devices web site address is www.moleculardevices.com.

It's also a good idea to test your hard disk and file structure when encountering problems. For problems not related to data acquisition, try more than one data file to see if your problem is due to a corrupted data file. If the problem is associated with one protocol file, that file may be corrupted; use the Acquire > New Protocol command to recreate the protocol. If the program is generating severe errors, use the utility in the Axon Laboratory folder, Reset to Program Defaults, to clear the registry entries. Please report any known bugs to Molecular Devices.

Context-sensitive menus are used. If you are unable to find or access a particular feature, it might be associated with a different type of window from the one that is active, i.e. that has its title bar highlighted. Or, a certain acquisition mode might need to be active for associated operations to be available. Sometimes, a certain parameter will need to be selected to enable additional parameters. The extensive online Help should be consulted for assistance.

HARDWARE PROBLEMS

If you have other equipment hooked up to your data acquisition system, start by disconnecting everything, except for a single BNC cable connecting Analog Out #0 to Analog In #0, to isolate the problem to the computer system. If you are still unable to determine if you have a hardware or a software problem, try moving your Digidata digitizer to another computer and see how it runs there. If it runs OK, then you have a computer-related problem.

Besides the data acquisition system itself, other common causes of intermittent hardware problems are defective cables. Try using different cables. Our technical support department will help you to further diagnose your problems.

SERVICE AND SUPPORT

Molecular Devices is committed to providing superior service and support for all of our products.

In order for us to diagnose your problem, we need enough information to duplicate your problem here. You should know the version of your copy of Clampex or Clampfit (found in Help > About), and of the operating system that you are using (found in the System control panel.) If you receive an error message, please write down the exact text of the message. Besides sending us step-by-step instructions on how to reproduce the error, including a copy of the protocol file and data file is extremely helpful and will greatly reduce the time needed to resolve your problem.

Contacting Technical Support

Email: support@moldev.com

Telephone: 408-747-1700
800-635-5577 (toll-free, US only)
Fax: (510) 675-6300

Mail: Molecular Devices Corporation
3280 Whipple Road
Union City, CA 94587
U.S.A.

C. Resources

PROGRAMS AND SOURCES

Several software programs complement pCLAMP by providing additional analysis and plotting functions. The pCLAMP Requirements page of Molecular Devices' web site has links to most of these companies' web sites for additional information. Note that support of the new pCLAMP 10 ABF 2.0 file format has not been verified with these companies.

AxoGraph (Molecular Devices) directly reads ABF data files for whole-cell, minis analysis and graphics on Macintosh computers.

DataAccess (Bruxton Corp) imports ABF files into Excel, Igor Pro, Origin and SigmaPlot.

DataView (Dr. Heitler, University of St. Andrews) directly reads ABF files for data analysis.

DATAPAC 2K2 (RUN Technologies) directly reads *.dat and ABF data files for spike train analysis.

Experimenter (DataWave Technologies) converts its data files to ABF files for use with pCLAMP's single-channel analysis.

Mini Analysis Program (Synaptosoft, Inc.) directly reads *.dat and ABF files for overlapping minis detection.

Origin (OriginLab Corp.) directly reads ABF data files for general analysis and graphing.

SigmaPlot (Systat Software Inc.) Electrophysiology Module directly reads ABF data files for general analysis and graphing.

Index

A

ABFInfo utility 6, 19
access resistance 10
acquisition
- definition of term 10
- gap-free 79
- modes 10

action potentials, aligning peaks 124
aliasing 15, 17
alpha exponential fitting functions 215
amplifier mode 10
- telegraphs 35

amplitudes
- resolution 18
- single-channel 97
- single-channel events 167

analog-to-digital conversion 16
Analysis window
- Clampex 32
- Clampfit 85
- cursors 32, 86

auto-correlation analysis 105
automatic data analysis in Clampfit 76
Average Traces command 91, 96
averaging (in Protocol Editor) 39
AxoClamp telegraphs 27, 34
Axon Binary Format (ABF) files 18
Axon FSP 19
Axon Layout File (ALF) files 20
Axon Text File (ATF) files 19
AxoScope 2

B

baseline
- adjusting 120
- correction 90
- definition of term 9
- drift 80, 168
- removing drift 123

baseline waveform protocol (LPT) 59
Bessel highpass filter (8-pole analog) 147
Bessel lowpass filter (8-pole) 134
beta fitting function 211
binomial distribution fitting function 212
bins, defining (tutorial) 126
Boltzmann fitting functions
- charge-voltage 212
- shifted 213
- standard 213
- Z-delta 214

Boxcar Smoothing filter 136
"brief" events 97, 167
- in single-channel recordings 168

burst analysis 105, 174–175
- Poisson Surprise 174
- single-channel 100
- specified interval 175

Butterworth lowpass filter (8-pole) 17, 137

C

Calibration Wizard 48
capacitance compensation, in telegraphs 34
cell-attached patch-clamp, single-channel recording 79
channel kinetics 79
channel, definition of term 9
Chebyshev lowpass filter (8-pole) 139

Chebyshev lowpass filter (8-pole) filter 17
Chebyshev transform 187–201
- fit speed 200
- fitting success 201
- goodness of fit 200

Clampex 82
- Analysis window 32
- configuring Digidata digitizers 29
- configuring to work with digitizer 28
- cursors 32
- Data File Index (DFI) window 32
- demo mode 29
- Lab Book window 32
- Membrane Test window 33
- new features 1–2
- Online Statistics window 34
- overriding parameters 36
- Pause-View button 84
- Results window 33
- Scope window 33
- setup tutorial 3
- Statistics window 34
- window types 31–34

Clampfit
- all points histograms 98
- Analysis window 85
- applying automatic analysis 76
- auto-correlation analysis 105
- automatic subtraction of control files 94
- averaging traces 96
- “brief” events 97
- burst analysis 105
- changing signal polarity 95
- conditioning data 90
- cross-correlation analysis 105
- cursors 86
- curve fitting 179–210
- Data File Index (DFI) window 86

Clampfit, continued
- Fetchan Events List mode 96
- filtering single-channel data 95, 97
- fitting functions 104–105
- fitting single-channel results 101
- for Fetchan users 94–98
- for pSTAT users 99–103
- graphs
  - generating 87
  - Graph window 87
  - scaling options 87
  - types 87
- histograms 99
- importing files 90
- Kolmogorov-Smirnov test analysis 105
- Lab Book window 88
- latency analysis 105
- latency measurement 98
- Layout window 88, 105
- new features 2
- nonstationary fluctuation analysis 105
- opening pSTAT events lists 90
- P(open) analysis 105
- peri-event analysis 105
- pulse averages 96
- QuB for single-channel analysis 103
- Results window 88
- returning to factory default values 76
- sampling theorem 15–16
- segmented averages 96
- statistical tests 127
- tutorials
  - I-V graph from cell-attached recording 109–110
  - I-V graph from whole-cell currents 107–109
  - quick-trace vs. trace plots from whole-cell currents 111
- variance-mean (V-M) analysis 105

Column Arithmetic dialog 113

command potential 12
comment tags 47
computer system requirements 25
Concatenate Files command 92
conditioning trains, see *pre-sweep trains*
Convert File dialog (Clampfit) 90
Convert to Frequency button 100
Create Data dialog 112
Create Graph command 87
cross-correlation analysis 105
current, inward and outward 12
cursors, Analysis window 32, 86
curve fitting (graphs) 87
custom fitting functions 205–206

D

data
- acquisition 43
- acquisition modes 10–11
- adjusting baseline 120
- adjusting stimulus waveform 110
- aliasing 17
- all points histograms 98
- amplitude (single-channel) 97
- amplitude resolution 18
- analog 16
- analog-to-digital 16
- analyzing in QuB 103
- averaging traces 91, 96
- baseline correction 90
- concatenating files 92
- conditioning in Clampfit 90
- display options 75, 86
- displaying stimulus waveform 107
- duplicating signals 119
- filtering 17, 90, 115, 117
- filtering single-channel in Clampfit 95, 97

data, continued
- fitting single-channel results 101
- fitting to dwell-time 101
- fitting tutorial 122
- gap-free acquisition 79
- header information 19
- histograms 99
- importing to Clampfit 90
- interpolation 116
- latency measurement 98
- maximum file size 21
- naming files 36
- noise in single-channel recording 80
- normalizing 171
- optimal acquisition 16–18
- oversampling 17
- plotting as differential values 95
- pulse averages 96
- reduction 116
- removing baseline drift 123
- renaming signals 118
- sampling 17
- segmented averages 96
- setting acquisition mode 37
- sorting 129
- subtracting control files 91, 94
- subtracting signals 118
- temporal resolution 17

Data File Index (DFI)
- DFI files 20
- window in Clampex 32
- window in Clampfit 86

decay, definition of term 9
Define Graphs dialog (event detection) 88
definitions 7–10
demo mode 29
depolarizing, definition of term 12

Digidata digitizers
configuring in Clampex 29
telegraph connections 27
digital inputs 27
digital outputs 27
digitizer, configuring Clampex to work with 28
Distribute Traces view 114
dongle 25
DongleFind 6
drug application system (DAS) 81
dwell-time data, fitting to 101

E

electrical interference filter 148–156
diagram 150
in tutorial 115
start-up transients 153
electrode resistance 10, 45
electronic valve controller 81
episodic stimulation mode 11
epochs 38
definition of term 7
evaluation of multicomponent signals (tutorial) 116–122
Event Monitor dialog 93
Event Statistics command 94
Event Viewer 94
events
“brief” 167
“brief”, in single-channel recordings 168
definition of term 9
detecting in Clampfit 92–94
fixed-length mode 82
fixed-length mode vs. high-speed oscilloscope mode 82
single-channel searches 92
template searches 92
threshold-based searches 92
events, continued
viewing statistics 94
experiments
baseline stage 50
defining in LPT Assistant 50
definition of term 9
protocols 50
running 50
exponential fitting functions
alpha 215
cumulative probability 215
log probability 215
power 216
probability 216
product 216
sloping baseline 216
standard 217
weighted 217
weighted/constrained 217
external tags 79

F

Fast Fourier Transform 157, 159
Fast Graph dialog 99
event analysis 88
Fetchan 94–98
Events List mode 96
file support pack (FSP) 19
files
format specifications 19
formats 18–20
importing to Clampfit 90
naming 36
filtering 17, 90
Bessel highpass filter (8-pole analog) 147
Bessel lowpass filter (8-pole) 134
Boxcar Smoothing filter 136

filtering, continued
- Butterworth lowpass filter (8-pole) 17, 137
- Chebyshev lowpass filter (8-pole) 17, 139
- data (tutorial) 117
- digital filter characteristics 133
- electrical interference filter 148–156
- end effects 133
- estimating rise time 134
- filters in telegraphs 34
- finite vs. infinite response 131–133
- Gaussian lowpass filter 141
- in Lab Bench 80
- in tutorial 115, 121
- MiniDigi 2
- Notch filter (2-pole) 142
- RC highpass filter (single pole) 147
- RC lowpass filter (8-pole) 145
- RC lowpass filter (single pole) 143
- single-channel event amplitudes 167

fitting 179–210
- automatic seeding 179
- Chebyshev transform 187–201
- custom functions 205–206
- failure 181
- fitting data (tutorial) 122
- function parameters 180
- functions in Clampfit 104–105
- Levenberg-Marquardt method 182–184
- maximum likelihood estimation 201–203
- methods 179
- minimization functions 206
- model comparison 203
- models 179
- multiple-term models 206
- normalized proportions 209
- numerical limitations 181
- Simplex method 184–186
- single-channel results 101

fitting, continued
- split clock data 179
- variable metric method 186–187
- weighting 207
- zero-shifting 209

fitting functions
- beta 211
- binomial distribution 212
- Boltzmann charge-voltage 212
- Boltzmann shifted 213
- Boltzmann standard 213
- Boltzmann Z-delta 214
- exponential alpha 215
- exponential cumulative probability 215
- exponential log probability 215
- exponential power 216
- exponential probability 216
- exponential product 216
- Gaussian 218
- Goldman-Hodgkin-Katz 218
- Goldman-Hodgkin-Katz extended 218
- Hill (4-Parameter Logistic) 219
- Hill steady-state 219
- Hodgkin-Huxley 214
- Langmuir-Hill 219
- Lineweaver-Burk 220
- logistic growth 220
- Lorentzian distribution 220
- Lorentzian power 221
- Lorentzian power 2 221
- Michaelis-Menten 222
- Nernst 222
- Poisson distribution 223
- polynomial 223
- sloping baseline exponential 216
- standard exponential 217
- standard parabolic 222
- standard straight line 224

fitting functions, continued
- straight line, origin at zero 223
- variance-mean parabolic 222
- voltage-dependent relaxation 224
- weighted exponential 217
- weighted/constrained exponential 217

fixed-length events mode 10, 83
- vs. high-speed oscilloscope mode 82

Force Values command 91

Fourier Series 157

frequency analysis (single-channel) 100

F-Test 127

function terms, maximum number 22

G

gap-free acquisition 79

gap-free mode 10

Gaussian function 218

Gaussian lowpass filter 141

Goldman-Hodgkin-Katz extended function 218

Goldman-Hodgkin-Katz function 218

Graph window (Clampfit) 87

graphs
- all points histograms 98
- converting to frequency 100
- creating (tutorial) 127
- creating histograms 87
- curve fitting 87
- Fast Graph dialog 99
- generating in Clampfit 87
- histograms 99
- I-V from cell-attached recording (tutorial) 109–110
- I-V from whole-cell currents (tutorial) 107–109
- linear scaling 87
- logarithmic scaling 87
- normalization 87

graphs, continued
- normalizing data 171
- quick-trace vs. trace plots from whole-cell currents (tutorial) 111
- relationship to Results window 87
- square root 87
- types available in Clampfit 87

H

high-speed oscilloscope mode 11
- vs. fixed-length events mode 82

Hill (4-Parameter Logistic) function 219

Hill steady-state function 219

histograms 87, 99
- all points 98
- creating 87
- normalized 172

history of pCLAMP 6

Hodgkin-Huxley fitting function 214

holding level, setting in I-V tutorial 68

hyperpolarizing, definition of term 12

I

idealized trace (single-channel) 101

input signals, setting up 36

interpolation 116

Interpolation dialog 117

isolated patch-clamps
- single-channel recording 81

I-V tutorial 65–77

J

Junction Potential Calculation (JPC) files 20

Junction Potential Calculator 48

K

Kolmogorov-Smirnov (K-S) test 169
  analysis 105

L

Lab Bench 35–36
  applying filtering 80
  overrides, in I-V tutorial 68
  telegraphs 35
Lab Book
  file format 20
Lab Book window
  displaying statistics 128
  in Clampex 32
  in Clampfit 88
Langmuir-Hill function 219
latency
  analysis 105
  analysis, single-channel 101
  measurement 98
Layout window
  in Clampfit 88, 105
  pasting waveforms into 111
  tutorial 111
level updating, in single-channel recordings 168
Levenberg-Marquardt method 182–184
  convergence 183
  precision 183
Lineweaver-Burk function 220
logistic growth function 220
long-term depression (LTD) 49
long-term potentiation (LTP) 49
Lorentzian distribution function 220
Lorentzian power 2 function 221
Lorentzian power function 221
LTD waveform protocol (LPT) 62
LTP Assistant 49
  alternating sweeps 55
  Baseline tab 50
  conditioning stage of experiments 54
  Conditioning tab 50
  defining waveforms 56
  first baseline stage of experiments 54
  input channels 52
  Inputs/Outputs tab 50
  output channels 51
  protocol configuration 55
  second baseline stage of experiments 54
  sequencing keys 54
  Sequencing tab 50
  statistics 63
LTP experiments, temporal structure 54

M

Mann-Whitney U-Test 127
math signals 41
maximum likelihood estimation 201–203
membrane resistance 46
Membrane Test 44–47, 79
  calculations 163–166
  in I-V tutorial 68
  Membrane Test window (Clampex) 33
  pulse train 45
  resistance 10, 163–166
  tutorial 77–78
Michaelis-Menten function 222
MiniDigi
  description 2
  filtering 2
  installing 29
  interface 2
  specifications 3
minimization functions 206

mode, acquisition (definition of term) 10
mode, amplifier (definition of term) 10
modes, data acquisition 10–11
Modify Signal Parameters command 118
MultiClamp, telegraphs 27, 34
multiple-term fitting models 206

N

Nernst function 222
new features
  Clampex 1–2
  Clampfit 2
nonstationary fluctuation analysis 105
normalization functions 170–172
normalization of graphs 87
Normalize command 171
Normalize Traces command 92, 170
Notch filter (2-pole) 142
Nyquist frequency 15

O

online help 4
Online Statistics window 34
oocyte recording 83
output signals, setting up 36
overrides 36
oversampling 17

P

P(open) 81
  analysis 105, 176
  single-channel analysis 100
P/N leak subtraction 42
parabolic function
  standard 222
  variance-mean 222
patch clamp, definition of term 13
patches
  cell-attached 13
  inside-out 13
  outside-out 14
Pause-View button 84
pCLAMP
  analyzing data with third-party software 19
  compatible 3rd-party software 233
  definitions of terms used in 7–10
  documentation 3
  file formats 18–20
  history 6
  installing software 27–28
  printer and plotter support 30
  resetting to defaults 30
  uninstalling software 28
  utility programs 6
peaks, definition of term 9
peri-event analysis 105, 175
  single-channel 101
piezo-electric switching device 81
pipette resistance 10
point, definition of term 8
Poisson distribution function 223
Poisson Surprise 100, 174
  burst analysis 174
polynomial function 223
postsynaptic pairing protocol (LPT) 62
Power Spectrum dialog 115
pre-sweep trains 42
  maximum pulses 22
Protocol Editor 37–43
  Input tab 39
  Math tab 41
  Output tab 39
  Statistics tab 41
  Stimulus tab 42

Protocol Editor, continued
Trigger tab 39
Update Preview button 42
User List 43
Waveform tab 41
see also *protocols*
Protocol File (PRO) files 20
protocols
acquisition mode 37
averaging 39
baseline waveform (LPT) 59
configuring in LTP Assistant 55
configuring waveforms 41
definition of term 9
epochs 38
in I-V tutorial 69–72
input channels 39
LTD waveform (LPT) 62
math signals 41
maximum number of epochs 21
output channels 39
P/N leak subtraction 42
postsynaptic pairing (LPT) 62
pre-sweep trains 42
sampling rate 38
setup tutorial 4
shape statistics 41
start-to-start interval 38
tetanus waveform (LPT) 60
theta waveform (LPT) 61
threshold-based statistics 40
trial hierarchy 38
trial length 37
triggers 40
pSTAT 99–103
events lists, opening in Clampfit 90
pulse averages 96
pulse train
in Membrane Test 45
setting number of pulses 78

Q

QuB, for single-channel analysis 103
Quick Graph command 87

R

RC highpass filter (single pole) 147
RC lowpass filter (8-pole) 145
RC lowpass filter (single pole) 143
Real Time Controls 43, 82, 84
telegraphs 35
reset to program defaults 6
resistance
access (definition of term) 10
electrode 45
electrode (definition of term) 10
membrane 46
membrane resistance, (definition of term) 10
seal 46
seal (definition of term) 10
series 46
total 46
total (definition of term) 10
Results File (RLT) files 20
Results window
basic statistics 126
Clampex 33
Clampfit 88
maximum rows 23
relationship to graphs 87
sorting data rows 129
Rich Text Format (RTF) files 20
rise, definition of term 9
Rolling display 83

runs
  definition of term 7
  maximum per trial 21

S

samples
  definition of term 7
  maximum per sweep 21
  maximum per trial 21
sampling interval, minimum 21
sampling rate 38
  maximum 21
  minimum 21
sampling theorem 15–16
Scale Factor Assistant 35
  in I-V tutorial 67
scale factors, telegraph 34
Scope window (Clampex) 33
  maximum number open windows 22
seal resistance 10, 46
seal test, see *Membrane Test*
search categories, maximum number 23
search commands, single-channel 97
Search Protocol File (SPF) files 20
Segmented Average command 91
segmented averages 96
separating action potentials by shape (tutorial) 123–129
Sequencing Key File (SKS) files 20
sequencing keys
  amplifier mode telegraphs 35
  in I-V tutorial 72
  in LTP Assistant 54
  maximum number defined 22
  Membrane Test 45
series resistance 10, 46
shape statistics 41
“short” events, see *“brief” events*
signals
  changing polarity in Clampfit 95
  configuration tutorial 4
  connections 26
  copying 85
  definition of term 9
  duplicating 119
  renaming 118
  subtracting 118
Simplex method 184–186
  convergence 185
  precision 186
single-channel recordings
  amplitude 97
  analysis in Clampfit 94–104
  “brief” events 168
  burst analysis 100
  calculating I-V 79
  cell-attached patch-clamp 79
  event amplitudes 167
  frequency analysis 100
  idealized trace 101
  isolated patch-clamps 81
  latency analysis 101
  noisy data 80
  P(open) analysis 100
  peri-event analysis 101
  preconditioning noisy (tutorial) 112–115
  search commands in Clampfit 97
  searches 92
  updating levels 168
software
  installing 27–28
  uninstalling 28
software protection key 25
specified interval burst analysis 175
square root (graphs) 87
standard straight line function 224

START, Trigger in 81
start-to-start interval 38
  minimum 21
start-up transients 153
statistics
  analyzing (tutorial) 125
  basic (in Results window) 126
  in Lab Book window 128
  in LTP Assistant 63
  shape 41
  statistical tests 127
  threshold-based 40, 79
Statistics File (STA) files 20
Statistics window (Clampex) 34
straight line function 223
Student's t-Test 127
subtracting control files 91, 94
sweeps
  alternating in LTP Assistant 55
  definition of term 7
  per run maximum 21

T

tags 47, 79
telegraphs 34–35
  amplifier mode 35
  amplifier mode sequencing keys 35
  AxoClamp amplifiers 34
  cable connections 27
  capacitance compensation 34
  configuration 35
  gain 34
  in I-V tutorial 68
  Lab Bench 35
  lowpass filter 34
  MultiClamp amplifiers 34
  Real Time Controls 35

telegraphs, continued
  setup tutorial 4
  signal scale factors 34
template matching 167
template searches 92
tetanus waveform protocol (LPT) 60
theta waveform protocol (LPT) 61
threshold-based searches 92
threshold-based statistics 40, 79
tooltips 4
total resistance 10, 46
traces
  definition of term 8
  idealized (single-channel) 101
Transfer Traces command (Clampfit) 89
transmembrane potential 12
trials
  definition of term 7
  hierarchy 38
  setting length 37
triggers
  Analog OUT signal as 82
  digitizer output 40
  in protocols 40
  Trigger In START 81
troubleshooting
  hardware 231
  software 231
tutorials
  analyzing statistics 125
  automatic data analysis in Clampfit 76
  channel and signal configuration 4
  Clampex setup 3
  creating graphs 127
  creating quick graphs 107–112
  defining bins 126
  evaluation of multicomponent signals 116–122
  I-V 65–77

tutorials, continued

I-V graph from cell-attached recording 109–110

I-V graph from whole-cell currents 107–109

Layout window 111

Membrane Test 77–78

preconditioning noisy single-channel recordings 112–115

protocol setup 4

quick-trace vs. trace plots from whole-cell currents 111

separating action potentials by shape 123–129

setting number of pulses in pulse train 78

statistical tests 127

telegraphs and cabling 4

U

unity gain 67

Update Preview button 42

User List 43

maximum characters in 22

utility programs 6

ABFInfo 6, 19

DongleFind utility program 6

reset to program defaults 6

V

variable metric method 186–187

convergence 186

precision 186

variable-length events mode 10, 83

variance-mean (V-M) analysis 105, 172–174

voice tags 47

voltage-dependent relaxation function 224

W

waveforms

adjusting stimulus waveform 110

defining in LPT Assistant 56

definition of term 7

displaying stimulus waveform 107

pasting in Layout window 111

whole-cell recording

evoked activity 83

oocyte 83

responses to drug applications 82

spontaneous activity 83

Z

zero-shifting 209