# pCLAMP Data Acquisition and Analysis Software

## User Guide

1-2500-0180 N
June 2025

MOLECULAR DEVICES

Patents: http://www.moleculardevices.com/patents

Product manufactured by Molecular Devices, LLC.
3860 N. First Street, San Jose, California, 95134, United States of America.
Molecular Devices, LLC is ISO 9001 registered.

# Contents


**Chapter 1: Introduction** ..... **9**

AxoScope Software ..... 9

MiniDigi Digitizer ..... 9

Clampex Software Quantitative Limits ..... 11

Clampfit Software Quantitative Limits ..... 12

File Formats ..... 13

Utility Programs ..... 15

Product Documentation ..... 15

About This Guide ..... 15

**Chapter 2: Concepts** ..... **17**

Software Interface Definitions ..... 17

Terms and Conventions in Electrophysiology ..... 20

The Sampling Theorem in pCLAMP Software ..... 23

Optimal Data Acquisition ..... 24

**Chapter 3: Setup** ..... **27**

Computer System Requirements ..... 27

Installing the Software ..... 31

Digitizer Configuration in Clampex Software ..... 34

**Chapter 4: Clampex Acquisition Software Features** ..... **37**

Clampex Software Windows ..... 37

Telegraphs ..... 40

Lab Bench ..... 40

Overrides ..... 41

Handling Data ..... 41

Edit Protocol ..... 42

Data Acquisition ..... 50

Real Time Controls ..... 50

Seal and Cell Quality: Membrane Test ..... 55

Time, Comment, and Voice Tags ..... 60

Junction Potential Calculator ..... 60

Calibration Wizard ..... 60

Sequencing Keys ..... 61

LTP Assistant ..... 61

**Chapter 5: Clampex Acquisition Software Tutorials** ..... **75**
I-V Tutorial ..... 75
Membrane Test Tutorial ..... 88
Scenarios ..... 89
**Chapter 6: Clampfit Analysis Software Features** ..... **95**
Clampfit Software Windows ..... 95
File Import ..... 100
Data Conditioning ..... 100
Event Detection ..... 102
Single-Channel Analysis ..... 104
Fitting and Statistical Analysis ..... 114
Creating Figures in the Layout Window ..... 115
**Chapter 7: Clampfit Analysis Software Tutorials** ..... **117**
Creating Quick Graphs ..... 118
Preconditioning Noisy Single-Channel Recordings ..... 120
Evaluation of Multicomponent Signals: Sensillar Potentials with Superimposed Action Potentials ..... 123
Separating Action Potentials by Their Shape ..... 129
**Chapter 8: Clampfit Software Basic Analysis** ..... **135**
Fourier Analysis ..... 135
The Power Spectrum ..... 139
Windowing ..... 140
Segment Overlapping ..... 140
Transform Length vs. Display Resolution ..... 141
Membrane Test ..... 141
Template Matching ..... 144
Single-Channel Event Amplitudes ..... 145
Kolmogorov-Smirnov Test ..... 146
Normalization Functions ..... 147
Variance-Mean (V-M) Analysis ..... 149
Burst Analysis ..... 151
Peri-Event Analysis ..... 152
P(Open) ..... 153
**Chapter 9: Clampfit Software Advanced Analysis** ..... **155**
Batch Analysis ..... 155
Minis Search ..... 160

Population Spike Search ..... 169
Action Potential Search ..... 176
**Chapter 10: Digital Filters ..... 183**
Digital Filter Characteristics ..... 184
End Effects ..... 184
Finite vs. Infinite Impulse Response Filters ..... 185
Bessel Lowpass Filter (8 Pole) Specifications ..... 186
Boxcar Smoothing Filter Specifications ..... 188
Butterworth Lowpass Filter (8 Pole) Specifications ..... 189
Chebyshev Lowpass Filter (8 Pole) Specifications ..... 191
Gaussian Lowpass Filter Specifications ..... 193
RC Lowpass Filter (Single Pole) Specifications ..... 195
RC Lowpass Filter (8 Pole) Specifications ..... 196
RC Highpass Filter (Single Pole) Specifications ..... 198
Bessel Highpass Filter (8-Pole Analog) Specifications ..... 199
Notch Filter (2 Pole) Specifications ..... 200
The Electrical Interference Filter ..... 201
**Chapter 11: Curve Fitting ..... 209**
The Levenberg-Marquardt Method ..... 212
The Simplex Method ..... 214
The Variable Metric Method ..... 216
The Chebyshev Transform ..... 217
Maximum Likelihood Estimation ..... 231
Model Comparison ..... 233
Defining a Custom Function ..... 235
Multiple-Term Fitting Models ..... 236
Minimization Functions ..... 236
Weighting ..... 237
Normalized Proportions ..... 239
Zero-shifting ..... 239
**Chapter 12: Fitting Functions ..... 241**
BETA Function ..... 241
Binomial ..... 242
Boltzmann, Charge-Voltage ..... 243
Boltzmann, Shifted ..... 243
Boltzmann, Standard ..... 244
Boltzmann, Z-Delta ..... 244

Current-Time Course (Hodgkin-Huxley) ..... 244
Exponential, Alpha ..... 245
Exponential, Cumulative Probability ..... 245
Exponential, Log Probability ..... 245
Exponential, Power ..... 246
Exponential, Probability ..... 246
Exponential, Product ..... 246
Exponential, Sloping Baseline ..... 246
Exponential, Standard ..... 247
Exponential, Weighted ..... 247
Exponential, Weighted/Constrained ..... 247
Gaussian ..... 248
Goldman-Hodgkin-Katz ..... 248
Goldman-Hodgkin-Katz, Extended ..... 249
Hill (4-parameter Logistic) ..... 249
Hill, Langmuir ..... 250
Hill, Steady State ..... 250
Lineweaver-Burk ..... 250
Logistic Growth ..... 251
Lorentzian Distribution ..... 251
Lorentzian Power 1 ..... 251
Lorentzian Power 2 ..... 252
Michaelis-Menten ..... 252
Nernst ..... 252
Parabola, Standard ..... 253
Parabola, Variance-Mean ..... 253
Poisson ..... 253
Polynomial ..... 254
Straight Line, Origin at Zero ..... 254
Straight Line, Standard ..... 254
Voltage-Dependent Relaxation ..... 254
Constants ..... 254

**Chapter 13: Troubleshooting ..... 255**

Software Problems ..... 255
Hardware Problems ..... 255
Obtaining Support ..... 256

**Appendix A: References** ..... **257**
Primary Sources ..... 257
Further Reading ..... 259
**Appendix B: Resources** ..... **261**
Programs and Sources ..... 261

# Chapter 1: Introduction


pCLAMP™ Data Acquisition and Analysis software is the data acquisition and analysis software suite from the Axon Instruments Conventional Electrophysiology product line of Molecular Devices. Designed for a variety of experiments, pCLAMP software has become the standard for electrophysiological experimentation and analysis. The flexibility of the pCLAMP software allows researchers to adapt it to many uses outside its traditional applications in electrophysiology.

The pCLAMP software suite contains the following:

- Clampex Software, for data acquisition and production of stimulus waveforms.
- Clampfit Software, for data analysis.
- AxoScope software, for background chart recording.

You can order the optional accessory MiniDigi™ 2-Channel Digitizer separately.

Clampex Software is a versatile and powerful software tool to acquire digitized data of all types. While excellent for the acquisition of patch-clamp data, it is not limited to measuring voltage-clamp or current-clamp responses. Use the Clampex Software 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 Software is powerful data analysis software with a wide variety of statistics, analyses, transforms, and layout tools for electrophysiological data.

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

## AxoScope Software

AxoScope software is a subset of Clampex Software. The HumSilencer™ Adaptive Noise Cancellation System is available in the AxoScope software. AxoScope software provides several continuous data acquisition modes, but has no episodic stimulation mode, which means that there is no capacity to generate stimulus waveforms. The Membrane Test is not included. Other advanced features in Clampex Software, such as the LTP Assistant, the Junction Potential Calculator, and instrument telegraphs are not included. With the optional accessory MiniDigi digitizer, you can use the AxoScope software as a background chart recorder running alongside Clampex Software during experiments.

## MiniDigi Digitizer

The optional accessory MiniDigi 2-Channel Digitizer is a low-noise, two-channel digitizer, designed to function with AxoScope software as a digital chart recorder. It has two independent, 16-bit analog inputs, each of which provides digitization at up to 1 kHz. The MiniDigi digitizer communicates with the host computer through a USB interface that also provides power to the digitizer.

**Note:** pCLAMP software version 11.x and higher does not support the MiniDigi Model 1A Digitizer. Contact Molecular Devices Technical Support to purchase a replacement MiniDigi Model 1B Digitizer. See support.moleculardevices.com/.

### Filtering

The MiniDigi digitizer uses your choice of minmax or analog-like filtering. 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 low-pass, anti-aliasing filter with a cutoff frequency one fifth of the sampling rate.

### Interface Description

The front panel of the MiniDigi digitizer has two BNC connectors for analog input channels 0 and 1. 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)**

| Item | Specification |
|---|---|
| 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 |
| Anti-alias filter (per channel) | Three-pole, 1.5 kHz Bessel |

### USB Interface

The MiniDigi digitizer has a low-power (< 100 mA), USB 1 device to interface with the computer.

## Clampex Software Quantitative Limits

**Table 1-2: Clampex Software Settings**

| Location | Parameter | Quantity |
|---|---|---|
| Protocol Editor > 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 (DD1550B, DD1550A, DD1550)<br>250 (DD1440A) |
| | Min. sampling rate (Hz) | 1 |
| | Min. sampling interval (µs) | 2 µs (DD1550B, DD1550A, DD1550)<br>4 µs (DD1440A) |
| | Min. start-to-start interval | Sweep length (for example 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 | 8 (DD1550B, DD1550A, DD1550)<br>16 (DD1440A)<br>15 if a Math signal or P/N leak subtraction enabled |
| Protocol Editor > Outputs tab | No. analog output channels | 8 (DD1550B, DD1550A, DD1550,)<br>4 (DD1440A) |
| Protocol Editor > Statistics tab | Max. samples in boxcar filter smoothing window | 21 (for example 10 on either side of each sample) |
| Protocol Editor > Inputs tab | Max. epochs | 50 |
| | Max. waveform channels | 8 (DD1550B, DD1550A, DD1550,)<br>4 (DD1440A) |
| Protocol Editor > Stim tab | Max. characters in User List | 512 (including commas; unlimited with Repeat) |
| | Max. Leak Subtraction sub-sweeps | 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 Software Quantitative Limits

**Table 1-3: Clampfit Software 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. independent variables in a custom function | 1 |
| | Max. points for functions containing a factorial term | 170 |
| 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 |

## File Formats

### Binary Data (.abf)

Clampex Software acquires and stores data as Axon Binary Format (.abf) files. 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 to apply certain mathematical operations to the data. When you save a data file from an Analysis window, you can save it in either integer format or floating-point format.

Clampex Software and pCLAMP software can read integer or floating point .abf files. All pCLAMP 11 programs read all previous versions of pCLAMP .abf data files. In addition, there are several third-party software packages that directly support this integer binary data file format. See Programs and Sources on page 261.

### Text Data (.atf)

Clampex Software can save data files as Axon Text File (.atf) files, which is an ASCII text format. You can import these files into spreadsheet, scientific analysis, and graphics programs, and can use word processor and text editor programs to edit them.

**Tip:** The .atf format is recommended to transfer data into other programs that do not support the .abf file format. The .atf files use more storage space than data stored in a binary format and do not include the full header information as in .abf files.

### Plain Data Files

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

### 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, downloadable from the online Molecular Devices Support Knowledge Base. The Axon FSP includes library routines written in C++, and supports the reading and writing of .abf files and .atf files.

Axon Instruments' long-standing role in electrophysiology data acquisition systems has led to the support of several third-party software applications for the analysis of pCLAMP software data. You can find programs suited for your particular needs.

To use Clampex Software version 6 binary data files in third-party analysis software:

1. Record the data with the Clampex Software program option configured to ensure data files and protocols are compatible with pCLAMP Software version 6.
2. Change the data file extension from .abf to .dat.

To use Clampex Software version 10.x binary data files in third-party analysis software:

- In Clampex Software version 10.x , export .abf files as version 9.

To use Clampex Software version 10.6 binary data files in third-party analysis software:

- pCLAMP software version 10.6 data files cannot be read by earlier versions of pCLAMP software version 10.x. Update the software to version 10.6.

### Other File Formats

pCLAMP software uses a number of specialized file formats for files generated in specific contexts, including:

- **Axon Layout File (.alf)** - For data saved from the pCLAMP software Layout window, where you arrange data for presentation. The .alf file format has changed in pCLAMP software and is not compatible with earlier versions.
- **Data File Index (.dfi)** - For files saved from the Data File Index window in both Clampex Software and pCLAMP software. The Data File Index window is a file management tool.
- **Junction Potential Calculation (.jpc)** - For files saved from the Clampex Software Junction Potential Calculator.
- **Rich Text Format (.rtf)** - This is a general-use text format not specific to pCLAMP. Lab Book files save as .rtf files.
- **Protocol File (.pro):** - For protocol settings in the Clampex Software Protocol Editor. These files are stored as .abf files, like the binary data files.
- **Results File (.rlt)** - For files you save from the Results window in the Clampex Software or the pCLAMP software. Any graphs you generate from Results window data also save in the .rlt file.
- **Sequencing Key File (.sks)** - For sequencing key sets you save from Configure > Sequencing Keys in Clampex Software. Sequencing keys allow you to link protocols and the Membrane Test to follow in predefined sequences.
- **Search Protocol File (.spf)** - For search configurations you save from the three pCLAMP software event detection searches (single-channel, threshold, and template).
- **Statistics File (.sta)** - For data from the Clampex Software Online Statistics window. If you open these files in the pCLAMP software, these files open both on a Statistics window and as text in the Results window.
- **Recovery of Minis Search (.rms)** - The analyzed Minis search data saves in this format.

## Utility Programs

Along with the three main software programs in the suite, pCLAMP software includes a number of utility applications. These are installed in the following path: ..\Program Files\Molecular Devices\pCLAMP 11\. Use Windows Explorer to open the pCLAMP 11 folder and double-click to run them. The utility programs are:

- **Reset to Program Defaults** (ClearRegistry.exe) - Run this program from the desktop: Start > Programs > Molecular Devices > pCLAMP 11 > Reset to Program Defaults. Use it when you encounter strange behavior in the program or when you want to set various Windows registry components back to the factory default settings.

  **CAUTION!** If you use the Clampfit Software Advanced Analysis Batch Analysis functionality:
  Do not clear the registry before noting the active database name and path in the Batch Analysis status bar. Clearing the registry resets the database connection to the default Batch Analysis database and you must use the Select Database option to reconnect to the database .

- **DongleFind** (DongleFind.exe) - This application checks the computer for Axon Instruments software security keys (dongles), including network keys. You can set the application to report information about the keys it finds.

## Product Documentation

The following related guides are available on the Molecular Devices Knowledge Base at mdc.custhelp.com:

- *pCLAMP Software Release Notes*
- *pCLAMP Data Acquisition and Analysis Software Quick Start Guide*
- *Setting Up Clampex for Data Acquisition* tutorial
- *Clampfit Batch Analysis User Guide*
- *Axon Binary File (ABF) Format User Guide*
- *Digidata® 1550B Low-Noise Data Acquisition System Quick Start Guide*
- *Digidata 1550B Low-Noise Data Acquisition System User Guide*
- *MiniDigi 2-Channel Digitizer Setup Guide*
- *The Axon Guide — Electrophysiology and Biophysics Laboratory Techniques*

In addition, the software includes Help files.

## About This Guide

This guide is intended for the Axon users using pCLAMP software and Axon instruments to describe the basic functionality of pCLAMP software for data acquisition and analysis in the study of electrophysiology.

The information in this guide is subject to change without notice. You should review the guide on the Molecular Devices Knowledge Base at support.moleculardevices.com/ for the most up-to-date information.

# Chapter 2: Concepts


This chapter contains terms used in the software interface, electrophysiology conventions, and optimal data acquisition information, to help you when you use the pCLAMP software.

## Software Interface Definitions

The following terms in the software are defined below:

- **Waveform** - 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 have digital outputs associated with it.
- **Epoch** - 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.
- **Sample** - The datum produced by one A/D (analog-to-digital) conversion or one D/A (digital-to-analog) conversion. In analysis contexts, samples can be referred to as points.
- **Sweep** - 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. See Figure 2-1 for illustration of the relationship between runs, sweeps, channels and trials.

  **Tip:** Sweeps were known as episodes in previous versions of the pCLAMP software.

- **Run** - 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.
- **Trial** - The data digitized from one or more runs, and saved as a single file. See Figure 2-1 for illustration of the relationship between runs, sweeps, channels and trials.
- **Trace** - A continuous set of data samples from a single input signal. When data displays as sweeps, each trace represents a sweep within that signal.
- **Point** - 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.
- **Channel** - A physical connection through which analog and digital signals are received or transmitted. Channels are identified in the pCLAMP software by the name of the digitizer port where connection is made, such as, Analog IN #0, Digital OUT #5.

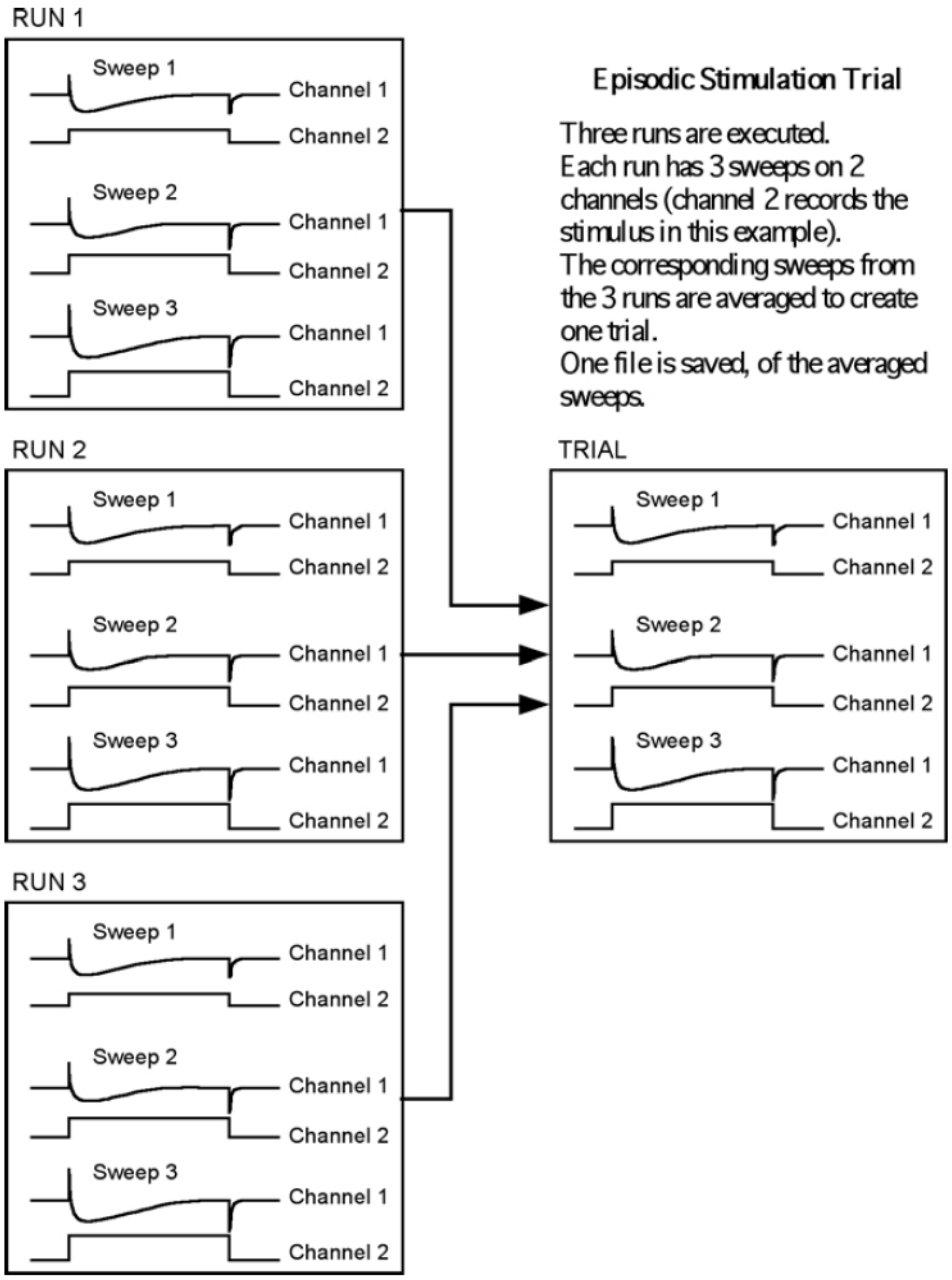



Figure 2-1: Clampex Software data structure showing relationship between runs and trial.

- **Signal** - A set of name, unit, scale factor, and offset, according to which:
  - Voltage inputs received at the analog input ports of the digitizer are represented in the Clampex Software as the physical parameter (unit) actually being read by the amplifier or transducer with correct scaling and offset, and name.
  - Voltage outputs generated through the analog output ports on the digitizer are represented in the Clampex Software as the physical parameter (unit) actually being delivered to the preparation by the amplifier or transducer with correct scaling, offset, and name.

  **Tip:** In the Clampex Software, you can configure numerous signals 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.

- **Protocol** - A set of configuration settings for a trial. It defines the acquisition mode, the trial hierarchy, for example the number of sweeps per run, and runs per trial, the sampling rate, the definition of the waveform, and other options, that can be saved into a *.pro protocol file.
- **Experiment** - Can be composed of several different protocols that 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.
- **Event** - 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.
- **Baseline** - In an episodic sweep that 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.
- **Peak** - Aa point in a trace of local maximum deviation from the baseline. Peaks can be positive (above the baseline) or negative (below the baseline).
- **Rise** - The 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.

  **Tip:** In previous versions of the pCLAMP software, rising phases were referred to with the term *left*, as in greatest left slope.

- **Decay** - The 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.

  **Tip:** In previous versions of the pCLAMP software, decay phases were sometimes referred to with the term *right*, as in greatest right slope.

- **Mode** - Determines how data acquisition is triggered and stopped, and whether it is accompanied by a command waveform. Most commonly referred to in the pCLAMP software as the *Acquisition Mode*, set in the Protocol Editor.

  Can also refer to *amplifier mode*, which is the amplifier state of current clamp, voltage clamp, or I = 0.
- **Acquisition** - The digitization of data by the Clampex Software. Acquired data can display in the Scope window at the same time they are recorded to disk or viewed without recording.

- **Electrode Resistance** ($R_e$) and **Pipette Resistance** ($R_p$) - The resistance due to the electrode. It does not include resistance due to environmental current-impeding factors near the electrode tip, such as, cellular debris, air bubbles, and poorly conducting solution.
- **Access Resistance** ($R_a$) - The sum of the electrode resistance and resistance due to current-impeding factors near the electrode tip, such as, cellular debris.

> **Tip:** Sometimes called Series Resistance ($R_s$), this is the term used on Axon-made amplifiers.

- **Membrane Resistance** ($R_m$) - The resistance across the cell membrane.
- **Total Resistance** ($R_t$) - The sum of membrane resistance and access resistance. When an electrode seals against the membrane, if the seal is successful. For example, a gigaohm seal access resistance is a negligible component of the total resistance so the total resistance is effectively equal to the seal resistance.
- **Seal Resistance** - The resistance given by the seal between the electrode tip and the cell membrane.

# Terms and Conventions in Electrophysiology

## Current and Voltage Conventions

When you use the Clampex Software to record membrane potential and current, there are various conflicting definitions of current and voltage polarities. The following topics describe the conventions used for Axon Instruments products. This information is presented in various instrument guides and other publications provided by Axon Instruments/Molecular Devices.

- **Positive Current** - The flow of positive ions out of the headstage into the micropipette and out of the micropipette tip into the preparation.
- **Inward Current** - Current that flows across the membrane, from the outside surface to the inside surface.
- **Outward Current** - Current that flows across the membrane, from the inside surface to the outside surface.
- **Positive Potential** - A positive voltage at the headstage input with respect to ground.
- **Transmembrane Potential** ($V_m$) - 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** - Depolarization is a positive shift in $V_m$ and conversely, hyperpolarization is a negative shift in $V_m$.

> **Tip:** 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 (for example, 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.

## Whole-Cell Voltage and Current Clamp

- **Depolarizing/Hyperpolarizing Commands** - A negative shift in the command voltage causes a negative shift in $V_m$ and is said to be hyperpolarizing. 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.
- **Transmembrane Potential vs. Command Potential** - The transmembrane potential is equivalent to the command potential. In whole-cell voltage clamp, the command potential controls the voltage at the tip of the intracellular voltage-recording micropipette.
- **Inward/Outward Current** - The current is always unambiguously defined with respect to the direction of flow into or out of the headstage.

**Note:** Some instrument designers 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.

In a cell generating an action potential, depolarization is caused by a flow of positive sodium or calcium ions into the cell. 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.



**Tip:** 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 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 always uses current and voltage conventions based on the instrument perspective.

## 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. The following explains these concepts further:

- **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

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 |

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 |

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 pCLAMP Software

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 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 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. The reconstructed 500 Hz waveform (heavy line) is only one-fifth the frequency of the original signal.

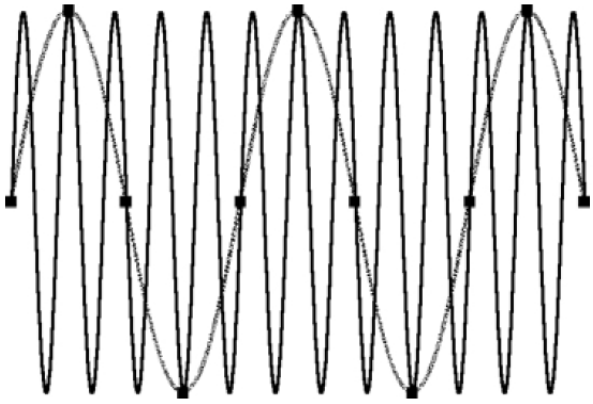


**Figure 2-2: Illustration of aliasing.**

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 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.

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 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 (for example 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, you 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 $1/\tau$ (where $\tau$ is the sampling interval), the signal must have no frequency components greater than $1/(2\tau)$. 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 who 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 your 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 (for example noise analysis, impedance analysis), it is common to over-sample only modestly. The main concern is to prevent aliasing. An anti-aliasing 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 anti-aliasing filter is ideal, for example, 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 (for example pulse analysis, IV curves), greater oversampling is required. This is because reconstruction of the analog signal requires not only an ideal anti-alias 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 (such as 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 value

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 you want to obtain the best resolution of the source signal, the need for amplifiers and/ or pre-amplifiers 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.

# Chapter 3: Setup

3

## Computer System Requirements

The pCLAMP software requires the following computer specifications:

**Computer System Requirements**

| Item | Description |
|---|---|
| Operating system | Windows 11, 64-bit<br>Windows 10, 64-bit |
| Memory | 8 GB RAM or more |
| Display | 1920 x 1080 display or higher |
| USB ports | 3 USB 2.0 ports |

### Software License Protection Keys

A software protection key device, commonly known as a dongle, enables the software for use. The key is a small USB device that plugs into the computer USB port. The type of USB key you are provided corresponds to the purchased software license. The available software license packages that require USB keys, include:

- pCLAMP Standard - includes Clampex Software (1 dongle)
- pCLAMP Advanced - includes Clampex Software and Advanced Clampfit Software (2 dongles)
- Clampfit Advanced Only - includes Advanced Clampfit Software (1 dongle)

**Identifying USB License Key Dongles**

| Software | Dongle Color | Part Number | |
|---|---|---|---|
| Clampex Software | Black | 5060221 |  |
| Advanced Clampfit Software | White | 5060223 |  |

If the key is not installed, Clampex Software runs in Demo mode, restricting you to simulated data.

## Signal Connections

The Clampex Software uses the following BNC connections on the Digidata 1550A digitizer series:

**Table 3-1: Clampex Software Digidata 1550A digitizer BNC Connections**

| Clampex Software Signals | Digidata 1550 BNC Sections | Digidata 1550 BNC Names |
|---|---|---|
| 8 Analog OUT Channels | ANALOG OUTPUTS (Front) | 0–7 |
| 8 Analog IN Channels | ANALOG INPUTS (Front) | 0–7 |
| 4 Telegraph Inputs (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 |

The Clampex Software uses the following BNC connections on the Digidata 1550 digitizer series:

**Table 3-2: Clampex Software Digidata 1550 digitizer BNC Connections**

| Clampex Software Signals | Digidata 1550 BNC Sections | Digidata 1550 BNC Names |
|---|---|---|
| 8 Analog OUT Channels | ANALOG OUTPUTS (Front) | 0–7 |
| 8 Analog IN Channels | ANALOG INPUTS (Front) | 0–7 |
| 4 Telegraph Inputs (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 |

The Clampex Software uses the following BNC connections on the Digidata 1440A digitizer series data acquisition systems:

**Table 3-3: Clampex Software Digidata 1440A BNC Connections**

| Clampex Software Signals | Digidata 1440A BNC Sections | Digidata 1440A 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 |

## Analog Output Signals

The Clampex Software 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 is connected via a BNC cable to a microelectrode current/voltage clamp amplifier External Command Input. If the amplifier has an internal command generator, be sure to switch to external command control. You can use the other analog output channels 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 the Clampex Software, these analog signals are digitized, display on the computer screen, and save to a data file on the hard disk.

## Digital Inputs

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

## Digital Outputs

The Clampex Software supports eight TTL-compatible digital outputs. These can be configured with bit patterns that coincide with the command waveform to allow you to 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. The Clampex Software can also output a dedicated scope trigger to synchronize signal digitization with an oscilloscope.

## Telegraphs

The Clampex Software can be configured to receive telegraphs from many amplifiers to report 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 1440A digitizer series and Digidata 1550 digitizer series have dedicated telegraph BNC ports. MultiClamp 700 and Axoclamp 900A amplifiers are computer-controlled so telegraphs are passed directly to the Clampex Software digitally with no telegraph cabling required. See Telegraphs on page 40.

## HumSilencer Adaptive Noise Cancellation System

The HumSilencer Adaptive Noise Cancellation System is integrated into Analog Input Channel #0 of the Digidata 1550A digitizer so no additional hardware installation is required. When it is in use, it eliminates electrical hum interference at 50 Hz or 60 Hz and the associated high-frequency harmonics. The ON/OFF switch for the HumSilencer system is software-controlled. The HumSilencer system works by learning the noise patterns that are synchronous with the AC power, known as hum, caching a signal-averaged replica of the hum, and subtracting the noise replica from the signal in real time. It adapts to noise changes in about 1 second within 50 power-cycles at 50 Hz or 60 Hz. It is designed to work with peak-to-peak hum amplitudes that, when combined with the signal of interest and other non-line-synchronous noise sources, are within the -10 to +10V range of the Digidata 1550A digitizer analog inputs. It is not a filter and does not have a filtering effect on acquired signals nor does it cause waveform distortion, such as, frequency change, amplitude attenuation, phase shift, or DC-voltage shift.

## Installing the Software

Before you start the pCLAMP software installer exit all other software programs, especially anti-virus software.

### Windows 7/10

Clampex Software version 11 runs on Windows 7 and Windows 10 (32- and 64-bit), for the Pro, Enterprise and Ultimate editions. The pCLAMP Setup program detects the operating system and loads the correct files.

**Note:** A Windows 10 (32-bit or 64-bit) operating system is recommended.

### First Time Installation

Disconnect all digitizers from the computer during the software installation.

To install the pCLAMP software suite for the first time:

1. Run the **Setup pCLAMP** program and follow the on-screen instructions.
2. In the Destination Folder dialog, you can change the destination drive and directory where the pCLAMP software is installed.
   The amount of hard disk space required for this installation and the amount of space available on the hard disk display. The default Molecular Devices Program Folder is created and program icons are added to it. You can rename the Program Folder or select an existing folders. Setup then copies the pCLAMP software files to the computer.
3. You should restart the computer for the Clampex Software to run properly.

### Updating the Software

Disconnect all digitizers from the computer during the software installation.

To install the pCLAMP software suite update:

1. Uninstall the older version of Clampex Software or AxoScope software using **Start** > **Control Panel** > **Programs** > **Uninstall a program**.
2. Install the pCLAMP software and then connect the digitizer.
3. Go to the **Start** menu and in the **Search** field enter **DEVICE MANAGER**.
4. In the Device Manager dialog, in the Molecular Devices folder, double-click the digitizer model number.
5. In the digitizer Properties dialog, select the **Driver** tab and click **Update Driver**.
6. When prompted, select **Search Automatically For Updated Driver Software**.
7. After the driver installs, verify that the new Driver Date is *5/16/2016* and the new Driver Version is *12.1.0.0*.

### Uninstalling the Software

This procedure works for previously installed AxoScope software and the pCLAMP software. The file locations are similar, but the folders have a different version number.

To uninstall the software:

1. Go to Windows **Start** > **All Programs** >**Molecular Devices**.
2. Open the folder for the software version to uninstall.
3. Select the version-appropriate Uninstall file.
4. Follow the procedures on-screen to uninstall the software.

## File Locations

File locations depend on the software version.

- **pCLAMP software version 11.x:**
  User-related files, such as data and parameter files, are stored in their own folders in:
  **\Documents and Settings\[user name]\My Documents\Molecular Devices\pCLAMP\...**
- **pCLAMP software versions 10.3 through 10.7:**
  System-related files, such as for the Lab Bench, System Lab Book, and user-defined telegraphs, are stored in the hidden folder:
  **C:\ProgramData\Molecular Devices\pCLAMP**
  Program application files are stored by default in the folder:
  **C:\Program Files\Molecular Devices\pCLAMP 10.7**
- **pCLAMP software version 10.2 and earlier:**
  Program application files and system-related files, such as for the Lab Bench, System Lab Book, and user-defined telegraphs, are stored in:
  **C:\Axon\pCLAMP X**

## Configuring MDC File Server

**Note:** The following procedure is for Advanced Analysis Clampfit Software only, which includes the Batch Analysis functionality. To run Advanced Analysis Clampfit Software, you must have the Advanced Analysis Clampfit Software USB dongle installed on the computer.

The MDC File Server runs on Windows 7 and 10 computer operating systems. The computer it runs on requires enough hard drive space to store the data files. You can change the storage location if it becomes full, and the database keeps track of data files in multiple storage locations. If you set a new storage location, keep the existing data storage location.

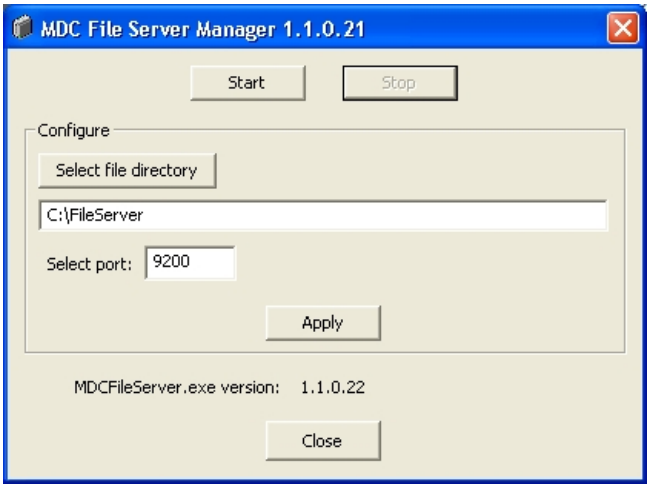
Before you start the Clampfit Software to run Batch Analysis the first time, you must first configure MDC File Server. MDC File Server is required to manage imported data. The MDC File Server must be running for Clampfit Software Batch Analysis to run.

To configure MDC File Server:

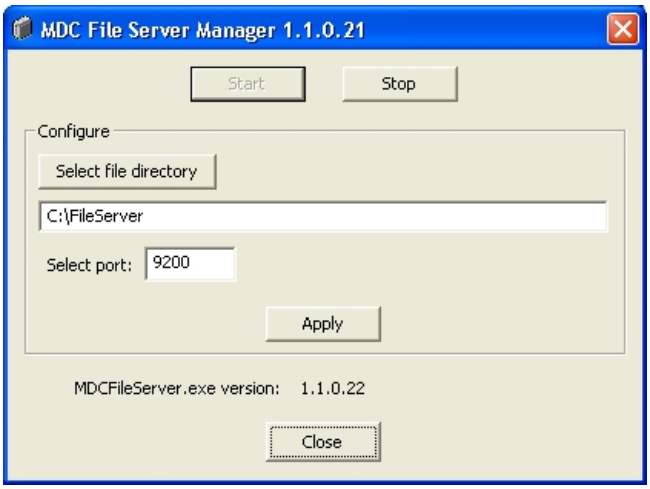
1. After you install the pCLAMP software, open the MDC File Server Manager from **Start** > **All Programs** > **Molecular Devices** > **MDC File Server** > **MDC File Server**, right-click and select **Run As Administrator**.

2. In the MDC File Server Manager dialog, click **Select File Directory** to display a Browse dialog where you select the folder in which to store data files.

**Note:** If you select a computer other than the local one, the computer must have access permissions to the folder location at all times.




3. Click **Create New Folder** and enter the folder name.
4. Click **OK**.
5. In the **Select Port** field, enter **9200**, and click **Apply**.
6. Click **Start** and wait for the Start button to become inactive.
7. When the Start button is inactive, click **Close**.




## Resetting Software Defaults

The Start > All Programs > Molecular Devices > pCLAMP 11 folder contains the utility Reset to Program Defaults, which resets the pCLAMP software 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:** Settings for other programs might display in the list of registry items, select the settings relevant to your situation.

## Digitizer Configuration in Clampex Software

After you install the pCLAMP software suite and connect the digitizer to the computer, you must configure the Clampex Software to communicate with the digitizer. This is done from the Configure > Digitizer dialog.

Detailed instructions on digitizer configuration are included in the *Digidata® 1550B Low-Noise Data Acquisition System plus HumSilencer™ Adaptive Noise Cancellation User Guide* and in the tutorial *Setting Up Clampex for Data Acquisition*, which is included in during the pCLAMP software installation in the following location:
**Start** > **All Programs** > **Molecular Devices** > **pCLAMP 11** > **Clampex 11 Tutorial**.

### Demo Mode

When you first install the Clampex Software it is in Demo mode and allows 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. From the Configure > Digitizer dialog with Demo selected, click **Configure** to alter the demo data output in non-episodic acquisition modes.

### Configuring the Digidata 1550B Digitizer in the Clampex Software

Connect the digitizer to the computer. If you have a Digidata 1550B digitizer, the Windows Found New Hardware Wizard displays. Follow the prompts until the installation completes. No separate driver is needed, so it is recommended that you search the hard drive for the driver.

**Note:** Configuring Digidata 1440A digitizer in the Clampex Software requires the same procedures, except the Digitizer Type selection varies.

To configure the Digidata 1550B digitizer in the Clampex Software:

1. Click **Start** > **All Programs** > **Molecular Devices** > **pCLAMP 11** > **Clampex Software 11** to start the Clampex Software.
2. In the Configure > Digitizer dialog click **Change**.
3. From the Change Digitizer dialog, select **Digitizer Type** > **Digidata 1550B Series**.
4. Click **Scan** to detect the digitizer. The first detected digitizer is assigned 0 and listed as Available. The Configuration changes from Not Present to reporting the selected digitizer model number, serial number, firmware version, HumSilencer channel availability (0, 1, or 4), and the OK button is active.
5. Click **OK** to exit the dialog, then click **OK** to exit the Digitizer dialog.

The front panel READY light is continuously on only when the software connects to the digitizer. After the Digidata 1550B digitizer warms up (allow one hour), it is ready to do experiments.

If you receive an error message, see the Troubleshooting chapter of the *Digidata® 1550B Low-Noise Data Acquisition System plus HumSilencer™ Adaptive Noise Cancellation User Guide*.

### HumSilencer Adaptive Noise Cancellation System

The HumSilencer system controls display only when connected to and configured with a HumSilencer Adaptive Noise Cancellation System-enabled Digidata 1550B digitizer. Not all Digidata 1550B digitizers have the HumSilencer Adaptive Noise Cancellation System (ANC). A *HumSilencer Options* sticker on the back of your digitizer specifies the number of ANC Inputs available.

**Note:** The HumSilencer Adaptive Noise Cancellation System controls do not display in the Configure > Digitizer > Demo configuration, nor is there a simulation mode for adaptive noise canceling.

When the Clampex Software starts, the HumSilencer system starts learning the line-frequency noise, until you disable the learning. See Setting Inputs and Outputs on page 45.

## MiniDigi Digitizer Installation

The optional MiniDigi digitizer works with the AxoScope software only. The Clampex Software does not support the MiniDigi digitizer.

To configure the MiniDigi digitizer in AxoScope software:

1. Run the pCLAMP 11 installer before you connect the MiniDigi digitizer to the computer.
2. After you install the pCLAMP, connect the USB cable to the USB port on the computer and to the MiniDigi digitizer.
3. In the Windows Found New Hardware wizard, follow the instructions until Windows installs the digitizer.
4. Click **Start** > **All Programs** > **Molecular Devices** > **pCLAMP 11** > **AxoScope 11** to start the AxoScope software.
5. Open the Configure > Digitizer dialog and click **Change**.
6. In the Change Digitizer dialog, select **MiniDigi** from the **Digitizer Type** field.
7. Click **Scan** to detect the digitizer. Available displays and **OK** is active.
8. Click **OK** to exit the dialog.
9. Click **Configure** to open the Configure MiniDigi dialog.
10. Select the style of filtering to use.
    - **Analog Filtering** - Applies a low-pass, anti-aliasing 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.
11. To calibrate the MiniDigi digitizer, attach a grounding plug to the Channel 0 BNC, and then click **Start**.
12. Repeat for Channel 1.
13. Click **OK**.

## Resetting Program Defaults

The Start > All Programs > Molecular Devices > pCLAMP 11 folder contains the utility Reset to Program Defaults, which resets the pCLAMP software settings back to the default values. Settings for other programs can display in the list of registry items. Select the items relevant to your situation.

## Printing

pCLAMP software supports printers and plotters you install through the Microsoft Windows operating system.

# Chapter 4: Clampex Acquisition Software Features


This chapter introduces the major features of the Clampex Software, including an extended discussion of the LTP Assistant. After you read this chapter, you can reinforce and extend your understanding by working through the tutorials in Clampex Acquisition Software Tutorials on page 75. More detailed information on the features discussed in this chapter is contained in the Clampex Software Help file.

## Clampex Software Windows

Clampex Software has a standard Windows format, with title bar, menus, and toolbars at the top, and a status bar along the bottom. In the View menu, you can choose which toolbars to display and select the buttons to appear in the toolbars using **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 you can drag it away to be repositioned like a standard dialog or attached to the right-hand side of the main window.

All commands available in the Clampex Software are included in the main menus. Many have toolbuttons or can be accessed from right-click menus. The main menu contents differ according to which of the windows is active.

Within the main Clampex Software window, you maximize, minimize, resize, and tile other windows. Right-click to display a menu with options specific to that window, including one that opens a Properties dialog. This allows you to configure the appearance of the window. You can save these settings as defaults using **View** > **Window Defaults**.

### Analysis Window

The Analysis window displays data that is saved in a file for review and measurement. The data displays graphically, as traces, with sub-windows for each signal stored within the file. Open a file in an Analysis window from **File** > **Open Data**.

Data can display as Sweeps or in Continuous or Concatenated modes, controlled from **View** > **Data Display**. In Sweep mode you can choose to view a 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 the sweeps by highlighting one at a time with the < and > keys.

Up to 24 cursor pairs, vertical, repositionable lines, are available to make simple measurements of the active sweep. Cursor text boxes can display 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 display the Cursor Properties dialog. A number of measurements for the first two sets of cursors, and the sections of trace they bound, can be sent to the Results window using the 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 parameters reported for each file. These options give you flexibility to organize and find files from experiments which is valuable to manage large amounts of data from archival sources. Create a Data File Index from **File** > **New Data File Index**.

## Lab Book

The Lab Book window is a text editor to log events that occur while the Clampex Software runs, for example when a protocol is opened, or when the holding level changes. Depending on the selection in **Configure** > **Lab Book Options**, the Clampex Software keeps a record of actions performed in the program, in the System Lab Book. You can open the System Lab Book window and type in it directly, or add comments to it with **Tools** > **Comment to Lab Book**, available when any window is active. Several tools offer the option to write values to the Lab Book, such as the Membrane Test.

The System Lab Book window is always open. Copies of the .rtf file can be saved for editing or archiving elsewhere.

## Membrane Test

Membrane Test is a utility to monitor a number of parameters during the three stages of a patch-clamp experiment:

1. **Bath** - Electrode resistance in the bath before patching, formerly called Seal Test
2. **Patch** - Patch resistance, to assist you in forming a gigaohm seal
3. **Cell** - Cell resistance and membrane capacitance

Use the Stage buttons at the top of the dialog to switch between the three stages. 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 to 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.

The Results window is open whenever the Clampex Software runs. 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 the pCLAMP software.

## Scope Window

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

Open multiple Scope windows with **Window** > **New Scope**. Use this to view 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) display with adjustable horizontal markers.

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

## Statistics Window

Statistics windows are graph-format windows with a time X axis and sub-windows for each of the statistical measurements recorded. Measurements from up to 24 different search regions (configurable within each sweep for High-Speed Oscilloscope and Episodic Stimulation acquisition modes) are color-coded within the separate sub-windows. A pane on the right-hand side of the window displays the search region color legend and reports the most recent data in numerical form.

When statistics are recorded in the Clampex Software, they are drawn in the Online Statistics window. This Statistics window becomes active and accepts data as soon as any statistics are generated. It continues to be active until you close the Clampex Software, 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 sub-windows are added to the window for each new type of statistics measurement.

When 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 resets the number of sub-windows to those enabled in the currently loaded protocol. Before you clear, you can save the contents of the Online Statistics window into a standard statistics .sta file. You can open these files into their own Statistics window in both the Clampex Software and the Clampfit Software 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).
- **Tools** > **Membrane Test** runs.

## Telegraphs

For many amplifiers, the Clampex Software 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 the Axoclamp 900A amplifier and the MultiClamp 700B amplifier telegraphing amplifier mode and signal scale factors and units. Gain telegraphs 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 amplifier and the MultiClamp 700B amplifier, amplifier mode changes can be linked to sequencing keys, so that, for example, the Clampex Software loads a protocol when you switch between voltage or current clamp modes. The scale factor and units telegraphs from the Axoclamp 900A and MultiClamp 700B amplifier almost entirely automate signal setup in the Clampex Software, leaving you only to name the input signals.

**Note:** Computer-controlled amplifiers have many the fields that are not available because the amplifier software takes direct control of these parameters.

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

1. If you use a computer-controlled amplifier such as the MultiClamp 700B amplifier or Axoclamp 900A amplifier, first turn on the amplifier and launch the corresponding Commander software.
2. Select the digitizer input channel on which to receive the signal relevant to the telegraphed information.
3. Select the amplifier type to determine the configuration options. If the amplifier does not support telegraphs, you can manually enter the information that would otherwise be telegraphed.
   - Select **Manual** as the telegraphed instrument in the Configuration dialog.
4. Enter the 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 reported in the Real Time Controls.

## Lab Bench

When you setup the Clampex Software for data acquisition, after you configure the digitizer, you must configure input and output signals for it. See Setup on page 27. This is done in the Lab Bench (**Configure** > **Lab Bench**). The Clampex Software allows you to define several different signals for each digitizer channel. For each signal you need to set units and scaling so that the Clampex Software displays data properly corrected for the parameter being read. When you set up an acquisition protocol you select the channels and signals from the Input and Output tabs in the Acquire > Edit Protocol dialog.

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

If you set 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 for non-computer-controlled amplifiers to:

- Define units of measurement and 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.
- Enter the settings on the Configure > Lab Bench > Input Signals tab. This section of the tab reports telegraphed values in the case of normal telegraphing. Telegraphed filter, gain, and capacitance compensation settings are reported in the Real Time Controls.

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

### Setting Up Output Signals

Use the Output Signals tab to:

- Define output signal units and scale factors using the Scale Factor Assistant
- Depending on the options you select in the Configure > Overrides dialog
  - Set the holding level
  - Set digital OUT channels

## Overrides

The Overrides dialog (**Configure** > **Overrides**) allows you to switch the control of various parameters, normally set in the Protocol Editor, to other locations. Most relevant are the options for analog and digital holding levels. If the Overrides options for these are left clear, they are defined, protocol by protocol, on the Outputs tab of the Protocol Editor. If selected, then levels are set on the Outputs tab in the Lab Bench. These levels then apply regardless of which protocol is run. Whichever location has control, 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 strategy for 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. You can select date-based naming or choose a prefix. Both options have long and short forms.

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

After you complete an experiment, you can inspect and edit the data by selecting **File** > **Last Recording** to open the most recently saved file in an Analysis window. The file uses the display defaults you set in the **File** > **Open Data** menu item.

### Save As

The original data file from a recording contains all data from all signals you configure in the protocol under which it was recorded. You can save sections from a file, define a region to save with the cursors, and select just the sweeps and signals to include.

To save selected sections:

1. Select **View** > **Window Properties** > **Show/Hide** to specify which signals remain visible.
2. Select **View** > **Select Sweeps** to specify which sweeps remain visible.
3. Select **File** > **Save As** > **Options** to change the Save As dialog settings.

## Edit Protocol

The Protocol Editor is important to the Clampex Software because it is where most experimental configuration occurs. Open with either the **Acquire** > **New Protocol** command or **Edit Protocol**, it has options to setup all aspects of data acquisition: the acquisition mode, trial length or hierarchy, sampling interval, the channels and signals to use, the shape of the command waveform, and whether or not to use adaptive noise cancellation, adapt between sweeps, triggering, statistics measurements, leak subtraction, pre-sweep trains, or math channels.

When you configure a complete range of acquisition settings, you can save them together as a protocol file (**Acquire** > **Save Protocol As**) and reopen it later (**Acquire** > **Open Protocol**).

### Setting the Mode/Rate

After you set up the hardware and configure signals in the Lab Bench, the first step to setup an experiment is to select an Acquisition Mode. See Data Acquisition on page 50.

In the Edit Protocol dialog, from the top section of the Mode/Rate tab, select an Acquisition Mode. See Data Acquisition Modes on page 44:

- Gap-Free
- Variable-Length Events
- Fixed-Length Events
- High-Speed Oscilloscope
- Episodic Stimulation

#### Trial Length

For Acquisition Modes other than Episodic Stimulation you must select options for Trial Length on the Mode/Rate tab. Options include:

- **Use Available Disk Space** - or until the data file reaches a 4 GB limit
- **Duration**

The amount of free space on the hard disk displays 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 is to use the **Acquisition Mode** > **Fixed-Length Events** or **Variable-Length Events**, instead of **Gap-Free**.

### Trial Hierarchy

When you select Episodic Stimulation the Mode/Rate tab has options for Trial Hierarchy. Enter the Trial Delay, Runs/Trial, Sweeps/Run, and Sweep Duration. If you have more than one input channel, the Sweep duration you enter is the same for each signal.

If you need to keep an eye on disk space, the file size for files created under the protocol at its current configuration displays beside the Sweeps/Run field. File size is unaffected by the number of runs per trial, each run in a trial is averaged and only one final, averaged run is saved. The total amount of free disk space displays at the bottom of the tab, both in megabytes and as the number of sweeps.

A breakdown of the sweep into first holding and last holding periods, and epochs is provided. This refers to the command waveform, where the first and last holding periods are calculated as 1/64th of the sweep duration, during which time only the holding level is output. The command waveform is output while recording data according to the hierarchy you configure. The waveform is produced in each sweep, and the epochs value that displays is the range of time available for you to configure its shape.

### Sampling Rate

A trial sampling rate is set on the Mode/Rate tab in the Edit Protocol dialog. The sampling rate is set per signal, so if you acquire multiple signals, each signal has the displayed sampling rate.

The total throughput of the data acquisition system, for example of [sampling rate] x [number of signals] displays beneath the sampling interval field for non-episodic modes, and at the bottom-right of the tab for Episodic Stimulation. This sampling rate is set per signal in the generation of the command waveform.

The Clampex Software supports split-clock acquisition so you can define two different sampling rates: a Fast rate and a Slow rate. You can specify when to use these rates in the Sample Rate row on the Waveform tab.

### Start-to-Start Intervals

**Mode/Rate** > **Episodic Stimulation** > **Start-to-Start Intervals** allows you to time the start of each sweep and/or run relative to the start of the one before it. A Start-to-Start Interval must be equal to or longer than the sweep length or the total time of a run. Pre-Sweep Train, P/N Leak Subtraction, Membrane Test Between Sweeps, and Adapt Between Sweeps are included in calculating the sweep length. To have the computer execute sweeps or runs as quickly as possible use the Minimum setting. The Minimum setting results in no lost data between sweeps to produce a continuous record divided into sweeps.

If you select a **Trigger** > **Trigger Source** to start a sweep, the Start-to-Start Intervals are not available because timing is controlled externally.

If you select **Stimulus** > **Adapt Between Sweeps** 1.081 seconds is added to Start-to-Start Intervals.

### Averaging

In **Mode/Rate** > **Episodic Stimulation**, Averaging is available when you specify more than one **Trial Hierarchy** > **Runs/Trial**. It is also 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 accumulate each successive run contributes a smaller percentage to the average, which becomes increasingly resistant to change. Use this option 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. Use this for data with significant baseline drift.

Up to 10,000 runs can be averaged. The average updates with every run. You can remove changes to the average which allows you to revert to an earlier saved average. If bad data are received midway through a trial, you can save some results by reverting to the last average before the undesirable data were recorded. Options for this are set in the **Averaging** > **Options**> **Undo File**.

### Data Acquisition Modes

The Clampex Software provides five 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. Use this mode 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. Use this mode to record 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 starts. 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.

Use Episodic Stimulation mode to study voltage-activated currents using the whole-cell patch-clamp configuration. For example, use the Clampex Software 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. Use online statistics to chart peak measurement values in real time.

## Setting Inputs and Outputs

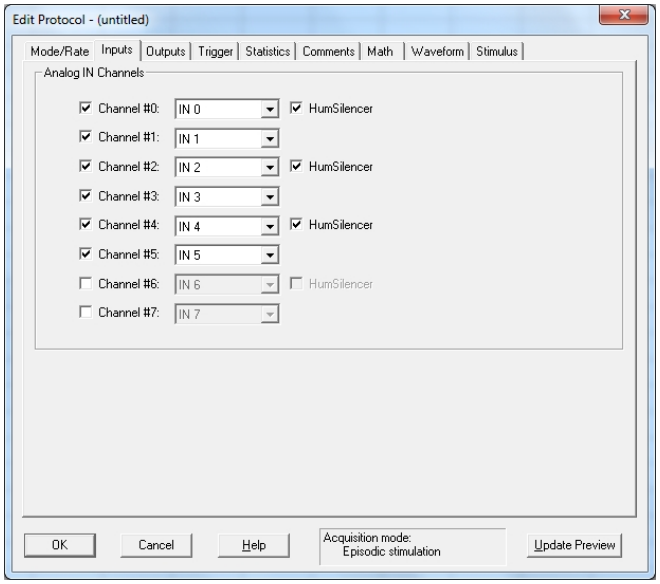
In the Edit Protocol dialog, on the Inputs tab, select the number of input channels to receive data and select the signals for each channel. These are the signals that you configure on the Configure > Lab Bench Inputs tab.

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

In setting analog holding levels, the levels you set in the Clampex Software are in addition to those set on the amplifier. Turn off the amplifier holding level control to control this entirely from the Clampex Software.

### Enabling the HumSilencer System Controls

When you use a digitizer with the HumSilencer adaptive noise cancellation system, either a Digidata 1550A digitizer or Digidata 1550B digitizer, to use the software-based controls, you must first turn them on in the software by selecting the **HumSilencer** checkbox on the Edit Protocol > Inputs tab for an Analog IN Channel (). When a checkbox is unavailable, the controls for the HumSilencer system in the Real Time Controls panel are not available.




After you enable the HumSilencer controls on the Edit Protocol > Inputs tab, you turn on and turn off the adaptive noise cancelling in the Real Time Controls panel by selecting or clearing the **HumSilencer** > **Subtract** checkbox. See Using HumSilencer Adaptive Noise Cancellation Controls on page 52.

**Note:** The Digidata 1550A digitizer provides the HumSilencer controls only for Analog IN Channel #0.
The Digidata 1550B digitizer provides the HumSilencer controls only for Analog IN Channels #0, #2, #4, and #6.

## Setting Triggers

Data acquisition triggering is set in the Edit Protocol dialog on the Trigger tab.

Your selection from the Start Trial With field determines the way you tell the Clampex Software you are ready for data viewing or recording. This is initiated by selecting **Acquire** > **Record or Acquire** > **View Only** (or clicking the equivalent icons the Acquisition toolbar). Thereafter, the option you select in Start Trial With takes effect.

The trial can be set to start immediately or on receipt of some external or keyboard signal. Rather than selecting a signal by name, you can go straight to the first signal enabled on the Input tab, with the selection of the First Acquired Signal option.

Once a trial starts, you can set the Clampex Software to await a trigger before data are recorded. In Acquisition modes other than Episodic Stimulation, you can select an input signal and then set a threshold level for it.

In all acquisition modes, except Gap-Free, on the Trigger tab of the Edit Protocol dialog, when the Trigger Source is set to **Digitizer START Input**, the **Timeout** field displays. Use it to set the maximum time the system polls for an external START trigger before it checks for user-intervention, such as clicking **Stop**, and responds to the intervention or resumes polling. Set this polling duration between 1000 ms to 60000 ms and make sure it is longer than the trigger interval.

The Pretrigger Length field allows you to set how much of the trace, before a threshold crossing, to record. Use this setting for the period that recording continues at the end of an event in Variable-Length Events mode. To avoid noisy signals causing false triggers, refine trigger settings in the Hysteresis dialog.

In Episodic Stimulation mode, Trigger Source > Internal Timer causes the runs and sweeps in the trial hierarchy to be controlled by the settings in Mode/Rate > Start-to-Start Interval.

You can enable a digital trigger signal from this tab. This uses the SCOPE front panel BNC on the Digidata 1550 digitizer and 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 use 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 select negative polarity) the threshold level you select in the Trigger Settings or Threshold-Based 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 you acquire in Gap-Free mode, Variable-Length Events mode, or Fixed-Length Events mode, you can monitor measurements such as the event frequency or percentage of time above the threshold level. These threshold-based statistics are set up on a protocol Trigger tab and use the same setting as 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 choose to have the data save in a statistics file when a recording finishes.

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

## Setting Statistics

Available in Episodic Stimulation mode and High-Speed Oscilloscope mode, Shape Statistics measure various parameters of evoked events such as peaks, areas, slopes, and rise times that you set on the Statistics tab in the Edit Protocol dialog.

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.

After 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 dragging the cursors to new positions.

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

## Setting 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 on the Math tab in the Edit Protocol dialog. Two signals can be scaled, offset, and arithmetically combined before being displayed by using **Equation** > **General Purpose**. You can use **Equation** > **Ratio Dyes** to measure cellular dye concentrations by ratioing the fluorescence signals from two photomultiplier tubes.

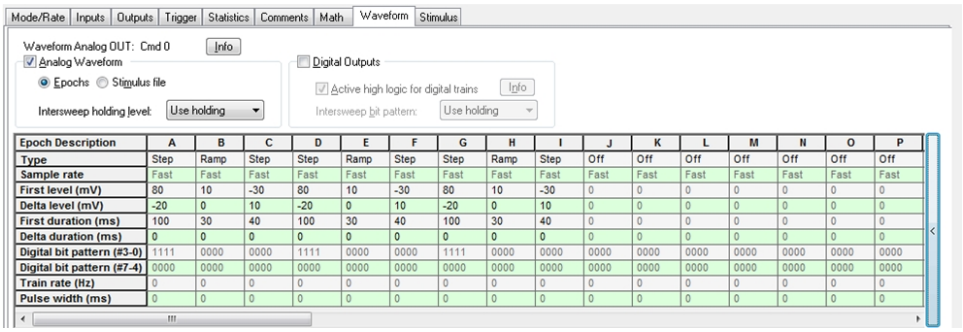
## Setting Waveform

The waveform outputs are available only in Episodic Stimulation mode. You can define analog and digital waveforms on the Waveform tab in the Edit Protocol dialog. For Analog waveform, you can configure each of the epochs to step to and hold a particular voltage, to increase or decrease linearly in a Ramp, to produce a Pulse train, Biphasic train, Triangle train, or Cosine train.

Configure train outputs for both analog and digital outputs. Trains can generate 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.

Up to eight analog stimulus waveforms can be generated simultaneously, one for each of the eight output channels tabs at the bottom of the dialog. Simultaneously, eight digital outputs can be enabled on one of the output channels.

Define epoch-driven waveforms in the Epoch Description table. The Epoch Description table consists of up to 50 Epochs, sections of the sweep, preceded and followed by a holding-level period that is predefined as 1/64 in duration of the sweep length. To configure an epoch, in the column you want to configure, click in the **Type** row, to assign an analog waveform type. Amplitudes and durations of each epoch can be systematically increased or decreased by setting delta values, for example, the fixed amounts of change in a parameter with every sweep.




| Epoch Description | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | Step | Ramp | Step | Step | Ramp | Step | Step | Ramp | Step | Off | Off | Off | Off | Off | Off | Off |
| Sample rate | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast | Fast |
| First level (mV) | 80 | 10 | -30 | 80 | 10 | -30 | 80 | 10 | -30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Delta level (mV) | -20 | 0 | 10 | -20 | 0 | 10 | -20 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| First duration (ms) | 100 | 30 | 40 | 100 | 30 | 40 | 100 | 30 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Delta duration (ms) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Digital bit pattern (#3-0) | 1111 | 0000 | 0000 | 1111 | 0000 | 0000 | 1111 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| Digital bit pattern (#7-4) | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| Train rate (Hz) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Pulse width (ms) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 4-1: Define up to 50 Epochs**

To copy and paste Epoch Description table columns:

1. Click in the data cell to copy, right-click and select **Copy**.
2. Click in the data cell into which to paste, right-click and select **Paste**.

To copy and paste a whole column of data:

1. Click the column number for the data to copy to highlight it.
2. Right-click and select **Copy**
3. Click the column number for the column in which to deposit the copied data to highlight it.
4. Right-click and select **Paste**.

For more flexible control of the analog waveform, see User List on page 49. If the epoch-based analog waveform is still not flexible enough, use the Stimulus file option to read data from an .abf or .atf file and output it as an analog waveform.

While you build the command waveform, click **Update Preview** in the bottom-right of the Edit Protocol dialog to open a Waveform Preview window that displays the waveform. Keep this window open while you make changes in the Epoch Description table. Click **Update Preview** to see the latest changes.

## Setting Stimulus

Available only in Episodic Stimulation mode.

### Pre-Sweep Trains

You can use a train of pulses to condition a cell prior to the main stimulus waveform. This is configured on the Stimulus tabs in the Edit Protocol dialog.

The Clampex Software pre-sweep train outputs analog pulses composed of repeated baseline and step levels. After the pre-sweep train completes, 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 generates, which is a convenient way to change the holding level for a 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 the Clampex Software. To view these, run the AxoScope software concurrently in Gap-Free mode with a MiniDigi digitizer.

### P/N Leak Subtraction

P/N Leak Subtraction is a technique used in voltage-clamp experiments to correct for a cell's passive membrane current, for example the leakage current. Configure this on the Stimulus tab in the Edit Protocol dialog.

In P/N Leak 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 runs.

For added flexibility to prevent 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. To prevent a conditioning response from the cell, the P/N waveforms can execute after the main stimulus waveform.

In the Clampex Software, both the raw data and the P/N corrected data are saved. The signal name for the corrected data channel is modified by a 'c' appending or replacing the last character of the signal name. During data acquisition, only one of these displays.

### User List

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

For example, rather than having sweep start-to-start times remain constant for every sweep in a run, set sweep start-to-start times 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, set independent epoch durations for each sweep. All aspects of the conditioning train, the number of subsweeps in P/N Leak Subtraction, and command waveform amplitudes and durations, besides other output features can be overridden from the User List.

### Membrane Test Between Sweeps

The Membrane Test Between Sweeps checkbox allows a Membrane Test to run between every sweep. In this mode the Membrane Test data are written to the Online Statistics window and display in the Real Time Controls.

### Adapt Between Sweeps

This feature is only available with a HumSilencer capable digitizer, either a Digidata 1550A digitizer or Digidata 1550B digitizer. Use this to allow inter-sweep learning while using the HumSilencer Adaptive Noise Cancellation System during long experiments to counteract the potential of noise pattern changes over time.

The **Adapt Between Sweeps** checkbox is available on the Stimulus tab in the Edit Protocol dialog during the configuration of an Episodic Stimulation acquisition protocol. The inter-sweep learning duration is 1.081 seconds. You must select at least one **Inputs** > **Analog IN Channel** available for HumSilencer and at least one **Waveform** > **Digital Outputs** channel.

**Note:** The Start-to-Start Intervals is calculated differently when you use Adapt Between Sweeps. The minimum time required to accommodate the epoch durations for a protocol includes the settings for Pre-sweep Train, P/N leak subtraction, Membrane Test Between Sweeps, and the 1.081 seconds time to adapt between sweeps.

## Data Acquisition

Use the Acquire menu or the toolbar buttons to set data acquisition options.

- You can digitize data and view the result in the Scope window, without saving anything to disk, with **Acquire** > **View Only**.
- After the acquisition completes, select **Write Last** to save it to disk. Click **Record** to save data saves to the disk during the acquisition.
- To overwrite the last recorded data file, click **Re-Record**.
- To repeatedly run the protocol, in either View Only or Record modes, click **Repeat**.

## Real Time Controls

The Real Time Controls panel on the left side of the main window below the top toolbar allows you to monitor the status of the experiment and to control input and output parameters while digitizing data (Figure 4-2).

You can test different experimental settings when you view live data without having to open the Edit Protocol dialog or Lab Bench dialog, 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 you select
- Pre-programming level, duration, and analog or digital values for Gap-Free mode
- Cancelling noise using HumSilencer system controls

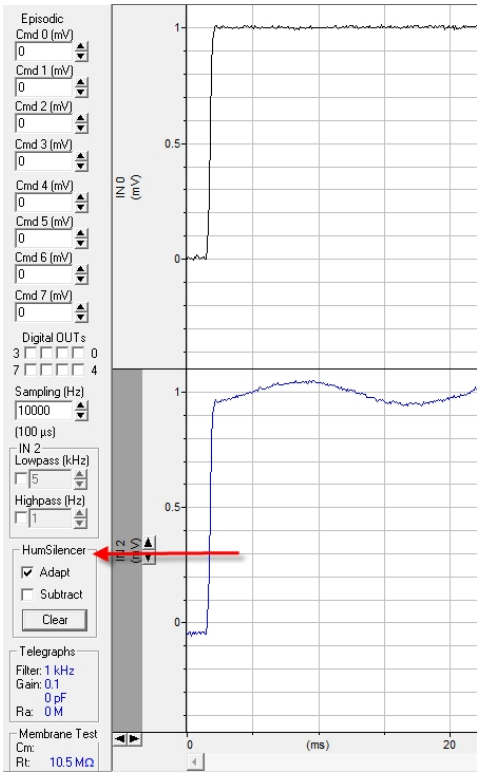



**Figure 4-2: Real Time Controls panel with HumSilencer system controls turned on**

The Real Time Controls panel 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 Leak Subtraction enabled.

To display the Real Time Controls panel:

- Select **View** > **Real Time Controls**.
  By default, the panel docks along the left frame of the main window.

To float the Real Time Controls panel:

- Click and drag the panel away from the from the left frame position.

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

- Most parameters are available when you acquire data in Acquire > View Only. Use this mode to experiment with sampling rates and filter settings.
- A few parameters are available when you record data in non-episodic modes. You can change analog and digital output levels and a comment tag is inserted into the data at each change.
- Most parameters are not available when you record data in Episodic Stimulation mode. You can change digital output levels and a comment tag is inserted into the data at each change.
- When you use Gap-Free mode, you can change epoch settings in the Real Time Controls panel or use the epoch description table. To access the table, click the **<** button.

## Using HumSilencer Adaptive Noise Cancellation Controls

When you use a digitizer with the HumSilencer Adaptive Noise Cancellation System, such as a Digidata 1550B digitizer, the controls for HumSilencer to learn real-time noise, activate noise cancellation, and reset noise-pattern learning, display in the Real Time Controls panel after you connect the HumSilencer-enabled digitizer to a computer and configure it with the pCLAMP software (Figure 4-2 and Figure 4-3).


**Figure 4-3: The HumSilencer System Real Time Controls in active learning mode.**

**Note:** The Humsilencer controls in the Real Time Controls panel do nothing until you turn them on with the Edit Protocol > Inputs > HumSilencer tab checkbox for an Analog IN Channel. See Enabling the HumSilencer System Controls on page 45.

The Real Time Controls for the HumSilencer system include:

- **Adapt** - Select this checkbox to turn on real-time noise pattern learning.
  Clear the **Adapt** checkbox to turn off the real-time noise pattern learning.

  **Note:** Clear the Adapt checkbox when there are big steps in the signal. Fast changes are visible as an excursion scaled to -1/50 of the fast signal amplitude in the output and takes ~1 second to age out of the rolling averaged noise replica. The Episodic Stimulation mode clears the Adapt checkbox.

- **Subtract** - Select this checkbox to turn on adaptive noise cancellation.
  Clear the **Subtract** checkbox to turn off adaptive noise cancellation.
- **Clear** - Click to reset the collected cache of the learned real-time noise patterns.
  HumSilencer Start and HumSilencer Stop comment tags display when a data file opens in an Analysis window (Figure 4-4).

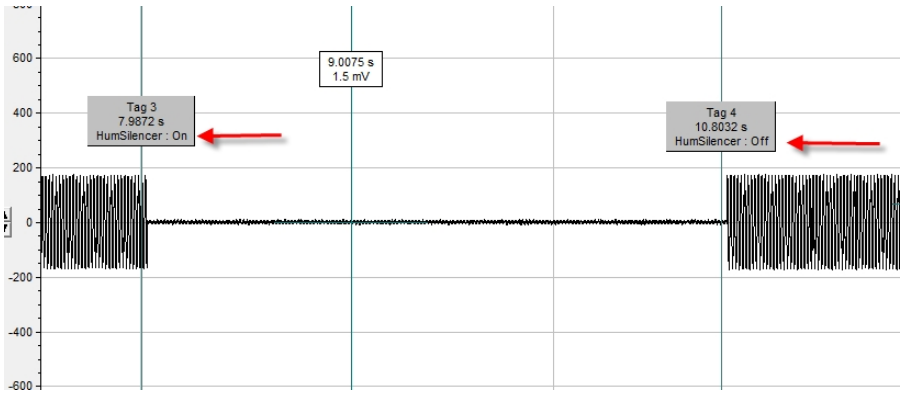



**Figure 4-4: HumSilencer Start and HumSilencer Stop comment tags in an open data file.**

The way you use these controls depends on the protocol data acquisition mode. See Setting the Mode/Rate on page 42.

**Note:** When you start a recording with HumSilencer adaptive noise cancellation turned on there can be a 21 ms recording-start delay while the HumSilencer system initializes.

For Gap-Free mode, Fixed-Length Events mode, High-Speed Oscilloscope mode, and Variable-Length Events mode:

1. Make sure to select an **Edit Protocol** > **Inputs** > **HumSilencer** tab checkbox for **Analog IN Channels**.
2. If not selected, in the Real Time Controls panel, select **Subtract** to turn on noise cancellation. The **Adapt** checkbox is automatically selected.
3. When you click **Record** or **View Only** to start data acquisition, you can change the HumSilencer control settings during the data acquisition.

For the Episodic Stimulation acquisition mode:

1. Make sure to select an **Edit Protocol** > **Inputs** > **HumSilencer** tab checkbox for **Analog IN Channels**.
2. If not selected, in the Real Time Controls panel, select **Subtract** to turn on noise cancellation. You can leave **Subtract** clear and noise cancellation is turned off.
3. When you click **Record** or **View Only** to start data acquisition, the **Adapt** checkbox is automatically cleared and the **Subtract** checkbox setting made in step 2 remains. During data acquisition, the HumSilencer control panel is not available to prevent setting changes.

**Note:** The HumSilencer system cancels noise during the time between episodic sweeps, including during the Edit Protocol > Stimulus tab functions: Pre-sweep Train, P/N Leak Subtraction, and Membrane Test Between Sweeps.

In all data acquisition modes, HumSilencer-modified data can display in a different color than the regular signal.

- Set the color from on the View > Window Properties > Signals tab in the Scope window (Figure 4-5).

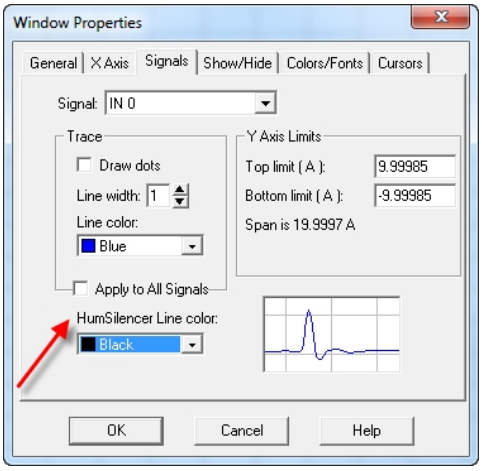



**Figure 4-5: HumSilencer line color change options in the View > Windows Properties > Signals tab**

## Pre-Program Gap-Free Mode Output

When you use Gap-Free mode in the Real Time Controls panel, click the **>** button to open the pre-programming dialog where you set the pre-program voltage level and holding duration values for each channel, as well as turn the digital bit on or off. You can pre-program up to 50 epochs. You can change values during a recording.

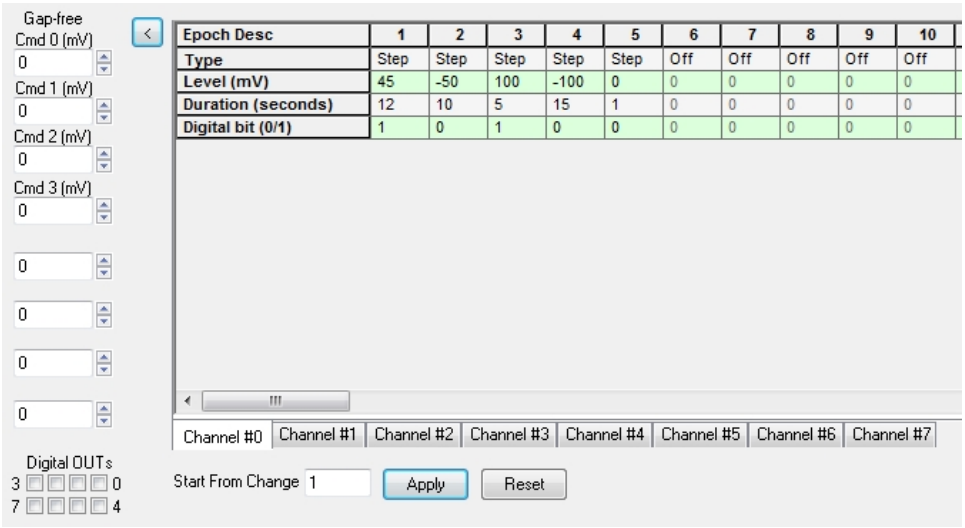



**Figure 4-6: Gap-free Mode Output Pre-Programming Dialog**

To pre-program the Gap-Free epochs:

1. In the **Epoch Desc** column number of choice (1 through 50), in the **Type** cell, where it is set to **Off**, click and select a holding pattern option such as **Step**.
2. Click in the **Level** cell and enter the voltage level in millivolts.
3. Click in the **Duration** cell and enter the holding time in seconds.
4. If you want this epoch to have digital output, click in the **Digital Bit** cell and enter **1** for on. By default Digital Bit is set to **0** for off.

   **Tip:** Each channel tab in the real time control panel represents one digital output channel on the digitizer.

5. Repeat steps 1 through 4.

   **Tip:** You can left-click on a cell and use the Copy and Paste options. To copy and paste a whole column, select a column by left-clicking on the column number, then when it is highlighted, right-click and select Copy or Paste.

6. In the **Start From** field, enter the **Epoch Desc** column number to specify which column settings to use first. By default, the setting is **1**.
7. Click **Apply** to save the settings or click **Reset** to clear the settings.
8. Select the **Channel #** tabs and repeat steps 1 through 7.

When the scope is active, you can apply changes. Whatever appears in the pre-program table, overrides whatever is set in the Controls panel.

## Seal and Cell Quality: Membrane Test

Tools > Membrane Test is a set of controls to monitor 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.

Membrane Test can run up to eight Analog Output channels and display multiple channels simultaneously. Independent control of voltage output at different stage configurations is enabled in each recorded cell.

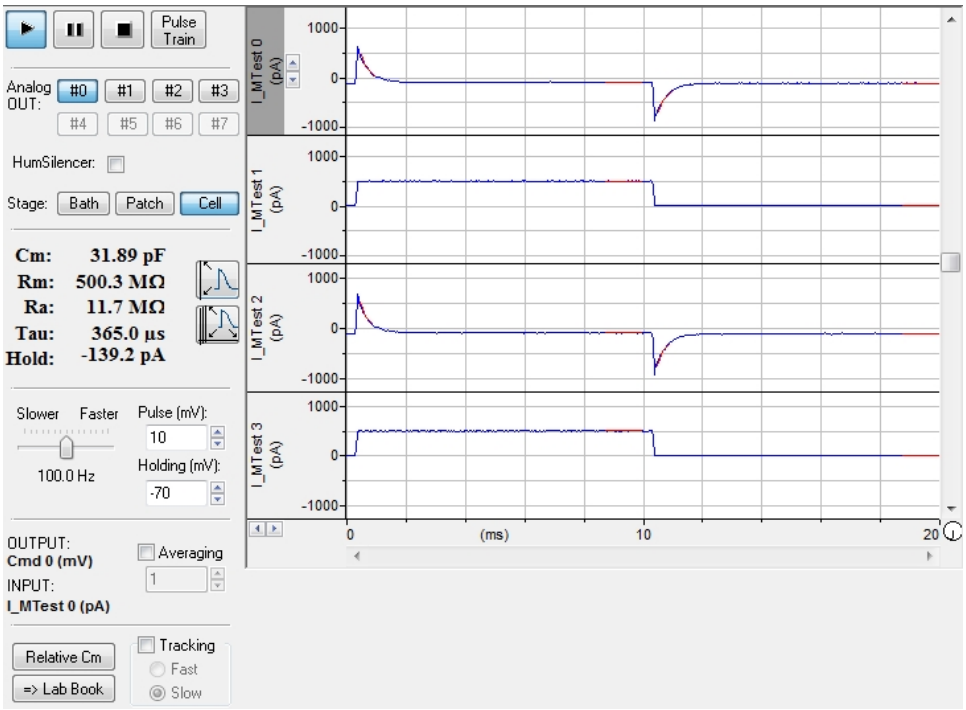



**Figure 4-7: Display of Multiple Membrane Test windows Simultaneously**

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

1. Bath stage monitors the electrode in the bath solution.
2. Patch stage monitors the creation of a gigaohm seal.
3. 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

The **Play**, **Pause** and **Stop** buttons allow you to control when the Membrane test runs, as well as its settings and test history. Membrane test can control the holding potential and stop generation of the test pulse. You can configure and trigger a pulse train to interrupt the test pulse.

Before you use a Membrane test you must first define input and output signals through Configure > Membrane Test Setup.

Membrane test measurements display in the Membrane Test window and can be charted in the Online Statistics window. You can save the test measurements as a .sta file with other statistics measurements from the Online Statistics window. A snapshot of the measurements at the current time point can be saved to the Lab Book.

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

Use the Membrane Test > Pulse Train to deliver a series of stimuli without closing the Membrane test (the test pulse ceases while the train is run). You can configure the pulse train 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. The fitted curve is used to calculate the transient peak, from which the access resistance is derived. The options in Configure > Membrane Test Setup allow you to define how much of the falling transient from each pulse edge to use for curve fitting. This dialog allows you to choose the timing and amplitude of the pulse train. You can enable and set up to four output channels.

In addition to measuring 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 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 73 and Membrane Test Tutorial on page 88.

## Using HumSilencer in Membrane Test

**Note:** There is no real-time noise learning during a Membrane test.

To use the HumSilencer adaptive noise cancellation system in a Membrane test:

1. In the Membrane Test dialog, select the **HumSilencer** checkbox to turn on the HumSilencer Real Time Controls (Figure 4-8).
2. In the Real Time Controls panel, select **Subtract** to turn on noise cancellation. Leave **Subtract** clear to turn off noise cancellation.
3. When you click **Record** or **View Only** to start data acquisition, the **Adapt** checkbox clears and the **Subtract** checkbox setting made in step 2 remains.
   During data acquisition the HumSilencer control panel is not available to prevent setting changes.

**Note:** When you start a recording with HumSilencer adaptive noise cancellation turned on there can be a 21 ms recording-start delay while the HumSilencer system initializes.

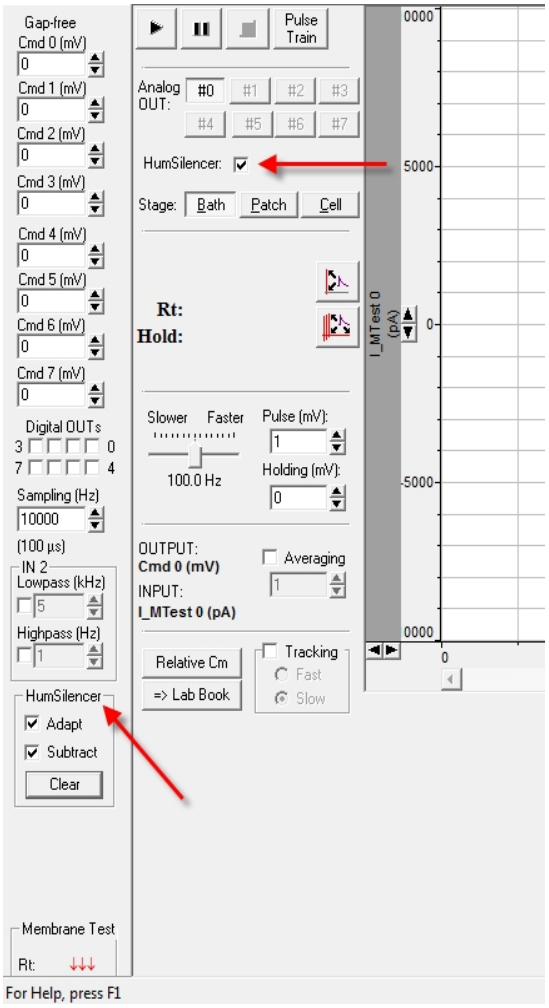



**Figure 4-8: HumSilencer system controls in the Real Time Controls panel and Membrane Test dialog**

## Membrane Test Measurements

In the pCLAMP software, the resistance due to an electrode alone is termed the electrode resistance ($R_e$). This is sometimes termed the pipette resistance ($R_p$). 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 (called $R_{debris}$).

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 termed series resistance ($R_s$). While Rs is the term used with Axon Instruments amplifiers, the pCLAMP software avoids its use as it can be confused with seal resistance.

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

The resistance between the headstage and ground is the total resistance ($R_t$). 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. The comparative resistances in the two pathways are such that one or the other pathway can be ignored, 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 4-9.

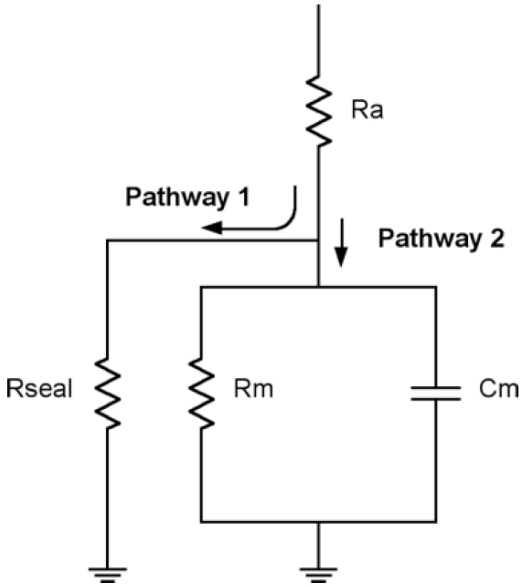



**Figure 4-9: Idealized circuit showing two pathways from electrode tip to ground.**

## 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. 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 termed the seal resistance. In this case total resistance 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 is achieved, the hope is that most current follows Pathway 2. In fact, with $R_m$ and $R_{seal}$ in parallel, these theoretically cannot be distinguished from each other. In this case, the software assumes 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 $\mathbf{R_s}$, which is used to indicate series resistance (for example, access resistance). For this reason the software avoids the use of $\mathbf{R_s}$, favoring instead $\mathbf{R_a}$, $\mathbf{R_t}$ or the words seal resistance or $\mathbf{R_{seal}}$ when it needs to refer to seal resistance specifically.

## Bath

When the electrode is in the bath the total resistance is 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 forms with the cell membrane is the reported value effectively a measure of seal resistance ($R_{seal}$).

The Membrane test allows you to control the holding potential which is a feature that is used to hyperpolarize the cell to aid in seal formation. This control remains only for as long as the Membrane test runs.

## Time, Comment, and Voice Tags

Time tags, Comment tags, and Voice tags allow you to annotate data while it is collected. These tags are available through the Acquire command during data acquisition and from toolbuttons.

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

If the computer has a sound card, you can insert Voice (audio) tags into data files. This is analogous to recording your voice on an audio channel when you save 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 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 allows you to define the experiment type and temperature, and the type of ground electrode to use.

The Junction Potential Calculator graphically displays the 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 the solution creates. Select the ions from the ion library list, or add your own ion definitions to the list. You can copy the results of the Junction Potential Calculator to the Lab Book, save them to disk, or print them.

For more information on how to use the Junction Potentials command, see the Help file.

## Calibration Wizard

Use the Calibration wizard to define the scaling and offset of a signal after the data is acquired. This is analogous to drawing a known scale bar on chart paper when you record continuous data to a pen recorder and then measure the scale bar to determine the scale factor for each signal of data. The Calibration wizard is available when you view data in the Analysis window and is activated by the **Tools** > **Calibration Wizard** command.

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

If you have defined the scale factor and offset in the Lab Bench you can update these settings in the data file using the Calibration wizard.

Existing data files can be rescaled in the pCLAMP software using Edit > Modify Signal Parameters.

## Sequencing Keys

The Sequencing Keys command in the Configure menu allows you to 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 define a tooltip to remind you what event is associated with each button.

In addition to setting various digital OUTs, the sequencing keys 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. It has been found that certain stimulus regimes alter synapse function, both to increase (LTP) and decrease (LTD) postsynaptic response. Tools > LTP Assistant provides an interface to setup LTP and LTD experiments.

The LTP Assistant dialog brings together a range of the Clampex Software functions needed for LTP and LTD experiments that are formatted to follow a sequence to setup these experiments.

The LTP Assistant dialog requires a basic knowledge of using the Clampex Software (for example, the Lab Bench). A range of pre-configured protocols are included along with default stimulation waveforms. Select the default protocol closest to your needs and adjust the settings or copy saved protocol files into the LTP Assistant.

If you want to design experiments with configuration options beyond those offered in the LTP Assistant dialog, you can use it for overall experiment management. For example, if the range of protocol definition options within the LTP Assistant dialog is insufficient, you can open the Edit Protocol dialog from within the LTP Assistant dialog (LTP Assistant > Baseline > Edit) and define the protocol with the broader range of options.

The LTP Assistant dialog has 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:

- **Sequencing** - General configuration of the stages that make up the experiment.
- **Inputs/Outputs** - Configuration of the Clampex Software 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. Enable default statistics measurements 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 in the LTP Assistant dialog is called an experiment. Experiments are saved on the Sequencing tab. This creates a folder that contains a sequencing key file (with the experiment's name and a .sks extension) and as many protocol files (.pro extensions) as you name on the Sequencing tab are incorporated into the experiment. These constituent files save at the same time as an experiment. Experiment folders save 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 on the Sequencing tab.

To run an experiment:

1. Open an experiment in the LTP Assistant dialog (Tools > LTP Assistant > Sequencing > Edit).
2. Close the LTP Assistant dialog to load the sequencing key file for the experiment into the Clampex Software.
3. Start the experiment by pressing the sequencing key for the first stage.
4. Subsequent stages can be started with sequencing keys, or may be configured to run when the previous stage finishes.

Each run of a protocol results in one data file, if you choose to record it. The data files generated during an experiment are named and located according to the Clampex Software functionality, set in >File > Set Data File Names. You can concatenate the files from an experiment in the pCLAMP software, with Analyze > Concatenate Files, provided they are recorded under similar conditions.

All statistics are written continuously to the Online Statistics window. You should clear the 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 save in one file. If 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), more configuration is 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 which are routes of neuronal activity across a synapse from a presynaptic to a postsynaptic cell. You can enable a maximum of two output channels for presynaptic stimulation. If you enable 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 leave one analog channel for postsynaptic stimulation. When two presynaptic outputs are enabled you can choose to alternate the baseline protocol delivery between these, sweep by sweep. This is activated on 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).

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 to 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. It is preferable to clamp recording electrodes from within the LTP Assistant dialog rather than with the amplifier because 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 to record or view 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 need to be clamped by the amplifier holding level control.

Alternatively, the experiment protocols can be 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 Software procedure. Create and configure signals in the Lab Bench in association with particular digitizer channels. On the LTP Assistant > Inputs/Outputs tab select specific signals for the channels you enable. This is the same as on the Input and Output tabs in the Edit Protocol dialog for standard protocols. If you are unfamiliar with the relation between signals and channels in the Clampex Software, see Software Interface Definitions on page 17.

## Example Setup

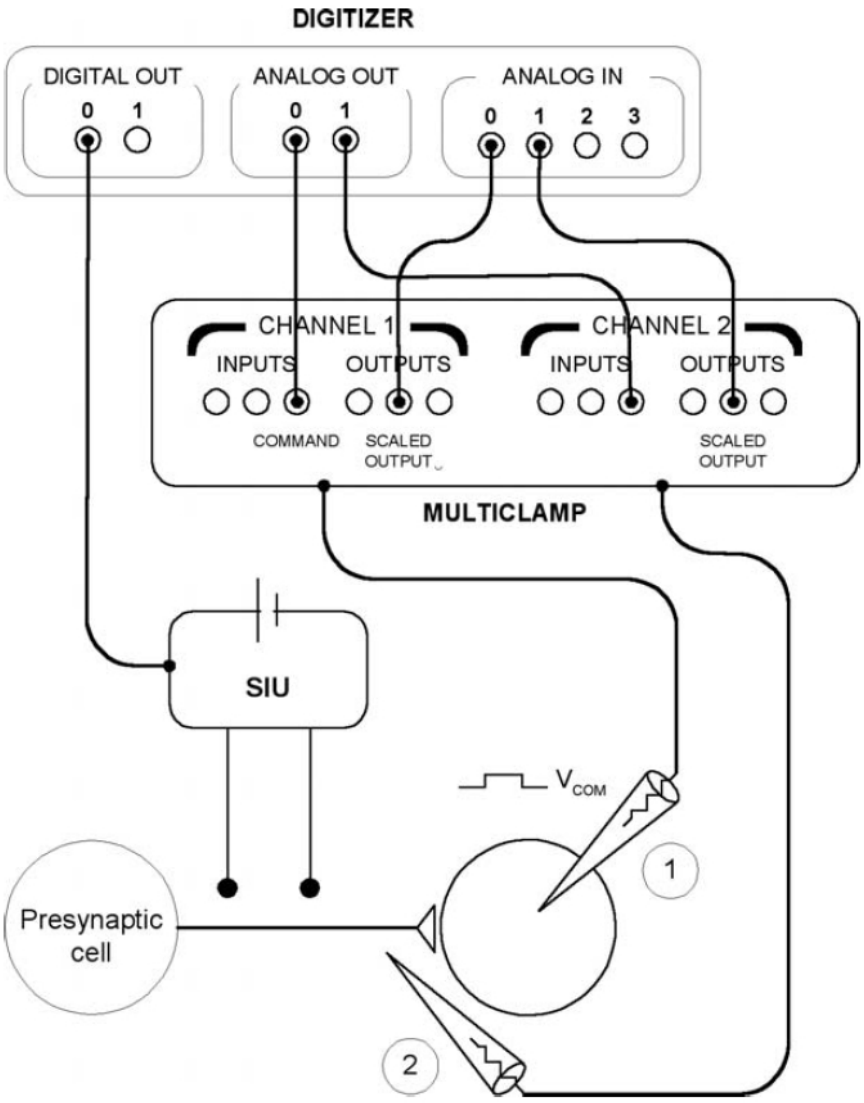



**Figure 4-10: Example of LTP experiment setup, with digital presynaptic stimulation, postsynaptic intracellular voltage step command and current recording, and extracellular recording.**

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 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, for example, 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. 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 displays on the Sequencing tab to which you can add as many stages as you like.

Each stage of the experiment is identified as either a baseline or conditioning protocol to determine the range of waveform configuration options available for that stage. The conditioning protocol is configured with default settings for tetanus, theta, or LTD stimuli. You can change these settings on the Baseline tabs or the Conditioning tabs. You can use a 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.

You can have as many baseline stages as you want, but they must use the same protocol and have the same stimulus waveform. 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 is view 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. This ordering of experiment stages does not control the order in which they are executed. Each stage is assigned a Start key and a Next key which is the Start key of the next stage to run. This allows you to have the stages run in any order by changing the Next key assignment in a protocol Properties. You can configure the completion of one stage to start the running of another stage, or manually interrupt a stage and start the next stage by pressing its sequencing key. You can have a stage link to itself so that it loops continuously until you manually start another stage.

The duration of each stage is set on the Sequencing tab. This can be set to the acquisition time you define in the protocol Sweeps section or you can override the protocol by setting a different time. This setting can cut short a protocol run 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 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. Conditioning stages are usually short and should be set to run for the protocol acquisition time calling the following baseline stage when finished.

### Protocols

The LTP Assistant dialog has one tab each for baseline and conditioning protocol configuration. These contain sub-sets of Clampex Software protocol-definition options. If you want more configuration options or you want to check the default settings not revealed on the LTP Assistant dialog, click **Edit** at the bottom of the tab to view the protocol in the 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 shown 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 post-waveform 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 postsynaptic pairing. You can view, record, and take peak measurements from the input signals while the waveform is being output (for example, 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 enable 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 Table 4-1

**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 |

Click **Copy From** on each protocol tab to select a protocol to copy into the LTP Assistant. Protocol settings are changed to be consistent with the settings current in the assistant at the time, for example, 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 settings.

Click **Preview** on the LTP Assistant to open an Analysis window that displays the waveform. This window updates 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.

Digital and analog waveform definition is essentially the same but 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.

The Clampex Software 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 displays in the Protocol Editor on the Waveform tabs. You can set the total duration of the epochs and the time available for waveform configuration in the Waveform Length field on each Baseline tab and Conditioning tab. 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 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. 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 4-11.

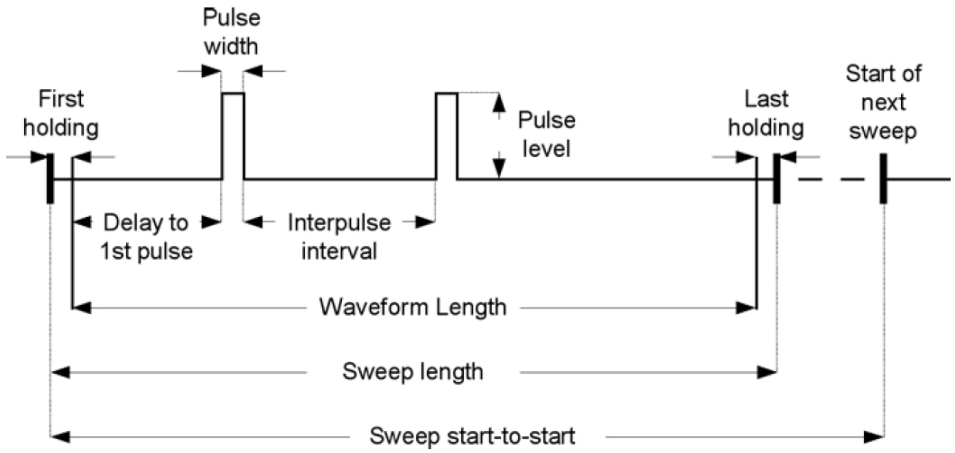



**Figure 4-11: Baseline sweep with two pulses per sweep showing configuration parameters used on Baseline tab.**

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 you enter in the Waveform Length field and the reported Sweep Length.

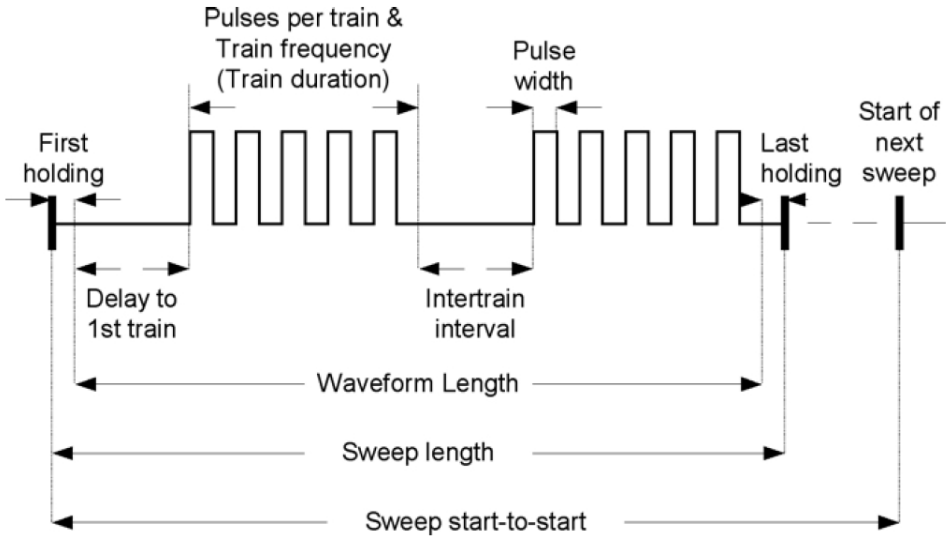



**Figure 4-12: Conditioning sweep configuration with two trains per sweep, showing parameters used on Conditioning tab.**

Because the pulses in a train each consist of a step and a following section at the holding level, trains 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 set.

## Predefined Protocols

Protocol definition does not start from a blank slate. When you outline the overall structure of an experiment on the Sequencing tab, default protocol settings load for each stage you create. One default protocol is used for all baseline stages and you have a choice of three default protocols for conditioning stages. This saves you time 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.

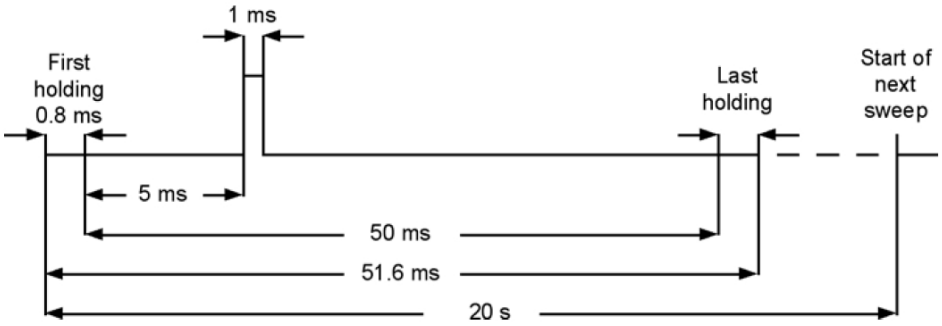



**Figure 4-13: Default baseline presynaptic waveform.**

- Sweep start-to-start: 20 s (0.05 Hz)
- Waveform length: 50 m
- 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 Add dialog on the Sequencing tab. Four 100 Hz, 1 s trains are delivered 20 s apart.

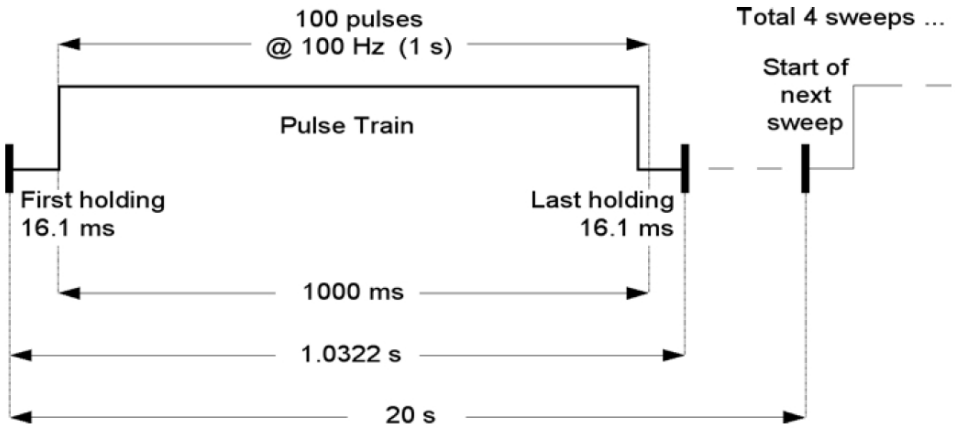



**Figure 4-14: Default Tetanus conditioning stimulus.**

- Sweep start-to-start: 20 s (0.05 Hz)
- Waveform length: 1000 ms
- Sweep length: 1.0322 s
- Pulses per train: 100
- Trains per sweep: 1
- Pulse level: 20 mV
- Delay to first train: 0 ms
- Train duration: 1000 ms
- Train frequency: 100 Hz
- Pulse width: 1 ms
- Intertrain interval: not applicable
- Sampling interval: 100 µs (10 kHz)
- No. of sweeps: 4

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

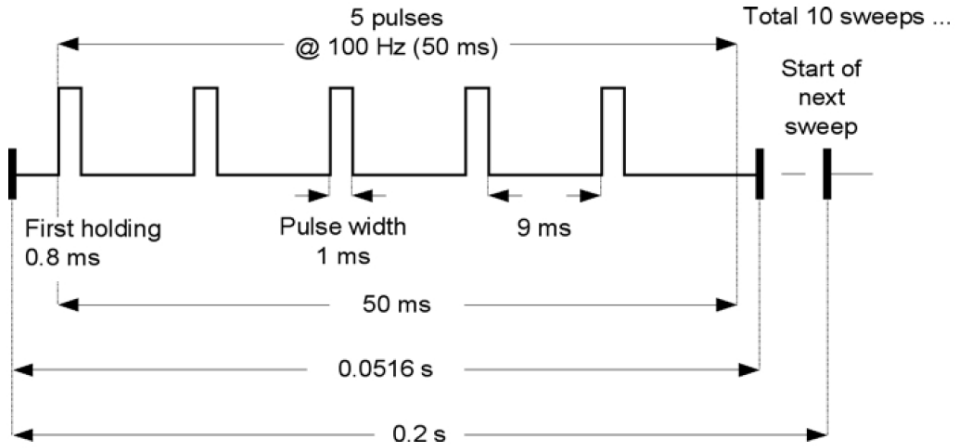



**Figure 4-15: Default Theta conditioning stimulus.**

- 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.

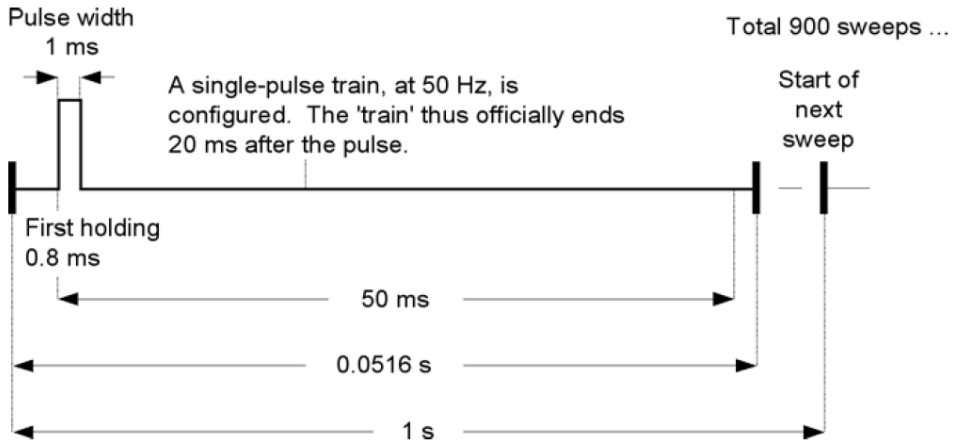



**Figure 4-16: Default LTD conditioning stimulus.**

- 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 requires that you enable a postsynaptic command channel 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 no output.

More often used in current-clamp experiments, you can 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, for example, digital OUT #0 for presynaptic stimulation, and analog OUT #0 for the postsynaptic command. You can pair postsynaptic stimulation with an analog presynaptic command, by enabling 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 is not calculated from the end of the presynaptic stimulus in this case.

## Membrane Test

To monitor access resistance over the length of an experiment, setup the experiment using Edit Protocol > Stimulus > Membrane Test Between Sweeps and sequencing keys.

Configure settling time following the end of the presynaptic stimulus before the Membrane test pulse generates to provide the cell some time to settle after stimulation from the presynaptic cell.

## Statistics

The LTP Assistant dialog allows you to record three online statistics measurements for all input signals during Baseline stages.

- Peak amplitude
- Peak time
- Slope

Enable these by selecting the **Statistics** checkbox for the channels on the Baseline tab. After you close the LTP Assistant dialog and start the experiment, set Search Regions in the Edit Protocol dialog, to capture the measurements properly.

Two different statistics, Search Region and Baseline Region, are used to measure these. Use Search Region 1 to measure peak amplitude and peak time and, with stable baseline data, drag the “1” cursor pair in the Scope window to capture the entire event evoked by the stimulus. Configure Search Region 2 to measure Slope, by linear regression for all data points within the region and to measure the Rise Slope of the event. Cursor pair “2” should 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 4-17):

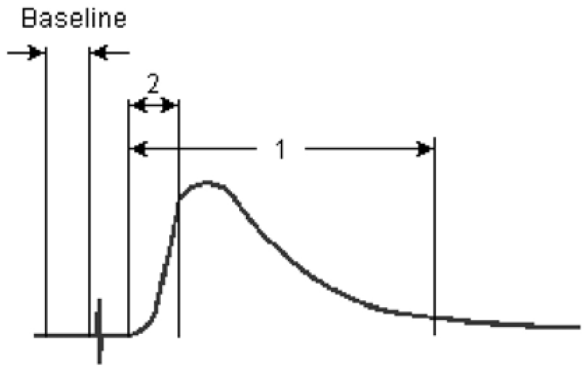



**Figure 4-17: Intended statistics baseline and search region positions.**

The regions you set save when the protocol closes which means that a protocol that opens after a conditioning stage retains the settings you made in the earlier stage.

Statistics measurements display continuously for the length of the experiment in the Online Statistics window. They can be saved as an .sta file 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, except with gaps where, for example, a conditioning protocol with no statistics recorded was run. If a very long baseline protocol is set and you stop the experiment manually, statistics do not save and you need to save them manually with **File** > **Save As**.

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

# Chapter 5: Clampex Acquisition Software Tutorials


This chapter presents information to help you perform experiments. It walks you through a step-by-step tutorial to set up experiments. It discusses how the Clampex Software can be used to further manipulate data files that have already been recorded. We discuss how to use the Quick Graph feature in pCLAMP software to plot an I-V (current vs. voltage) curve while the Clampex Software acquires data. It finishes by presenting several scenarios to do different types of experiments.

## I-V Tutorial

This tutorial is designed to walk you through the steps of setting up various types of experiments in the Clampex Software. The goal of this section is to teach you how to set up the Clampex Software to perform an experiment, rather than discuss each feature of the Clampex Software in detail. You may want to first follow the tutorial *Setting Up Clampex Software for Data Acquisition*, which covers the basic setup procedures for a Digidata digitizer along with the Axopatch 200B, Axoclamp 900A and MultiClamp 700B amplifiers. See Product Documentation on page 15.

**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:

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

Hardware used:

- Axopatch 200B amplifier, headstage set to $\beta = 0.1$
- Electronically controlled solution changer
- Digidata 1550 digitizer

### Selecting Digitizer

One of the first steps to perform an experiment is to select the hardware for acquisition. This is done with the **Configure** > **Digitizer** menu. By default, Clampex Software is installed with the demo digitizer active. This example uses the demo digitizer but you must select and configure the digitizer before you do 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 this tutorial example, an Axopatch 200B amplifier is used in voltage-clamp mode to record membrane currents in response to square voltage steps. This example sets up the Clampex Software to acquire the membrane currents on Analog IN #0, and to output the voltage command on Analog OUT #0.

## Setting Up Input Signals

To set up the Lab Bench:

1. Select **Configure** > **Lab Bench**.
2. On the Input Signals tab, in the **Digitizer Channels** list, select **Analog IN #0**.
   Notice that there are several default Signals associated with the channel.
3. Click **Add** to add another signal to Digitizer Channel > Analog IN #0.
4. In the **Add Signal** > **New Name** field, enter **Im** and click **OK**.
5. In **Scaling**>**Signal Units**, select the unit prefix **p**, and leave the units value set to **A**.

   **Tip:** For most amplifiers, there is no need to adjust the Offset, so leave the Offset field value at the default 0.

6. For the **Scale Factor** value, click **Scale Factor Assistant**.
   You need to know the gain settings of the amplifier, which can be read off of the front of the instrument. In this example, the signal is membrane current and it is assumed that the Axopatch 200B amplifier is set to an α value of 1 and a β value of 0.1.
7. In the Scale Factor Assistant, click **Yes**, select **Axopatch 200 Series**, which indicates the amplifier is connected to Analog IN #0, then click **Next**.
8. In **Mode Setting** , select **V-Clamp** for voltage-clamp mode.
9. In **Config Setting**, select **Whole Cell (β = 0.1)** to match the β setting of the Axopatch 200B amplifier.
10. In **Preferred Signal Units**, select **pA**.
11. For **Gain**, set the current output gain value to **1** to match the α setting of the Axopatch 200B amplifier.
    This is referred to as unity gain. The setting of unity gain is very important when you enable telegraphs, a functionality that is discussed in the next section. See Telegraphs, see page 78.

    For now, be sure to set **Gain** to **1**, and set up the Lab Bench to reflect this.

    The next section discusses how the Clampex Software handles changing the gain in the telegraph section.
12. Click **Finish**.
    $1e^{-4}$ *V/pA* computes and displays in the **Scale Factor** field.
13. In **Software RC Filter**, for this example, leave the **Lowpass** and **Highpass** filter value options off.
    You have finished setting up the Analog IN #0 channel to record *Im* signals from the Axopatch 200B amplifier. 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, otherwise you may want to have an additional gain applied to the signal before it is digitized, to ensure that you are maximizing the dynamic range of the digitizer.
14. Now configure **Analog IN #1** to record the membrane potential.
15. Add the 10 Vm OUTPUT signal from the Axopatch 200B amplifier to **Analog IN #1**.
16. In **Scaling**>**Signal Units**, select the unit prefix **m**, and set the units value to **V** to record voltage in *mV*.

17. Set the **Scale Factor (V/mV)** to **0.01** for the signal $V_m$ since the gain of the 10 $V_m$ OUTPUT of the Axopatch 200B amplifier is x 10.

    **Note:** Do not use the Scale Factor Assistant for this signal, because it assumes that the Analog IN channel is connected to the scaled output of the Axopatch 200B amplifier, and not to other signals such as 10 Vm.

18. You need to consider whether, with a gain of 10, the signal that is recording adequately uses the dynamic range of the digitizer.
    Typically, the cellular membrane potential is in the range of ±100 mV, and with a gain of 10, the Axopatch 200B amplifier outputs a voltage in the range of ±1 V (100 mV x 10).
    Given that the dynamic range of the digitizer is ±10 V, it is apparent that only ~10% of the dynamic range is used (±1 V/±10 V). But, for a 16-bit digitizer such as the Digidata 1550A digitizer, while not optimal, this is resolved into about 6,500 steps, which is still 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.

To set up the Lab Bench for this example:

1. Select **Configure** > **Lab Bench**.
2. On the Output Signals tab, in the **Digitizer Channels** list, select **Analog IN #0**.
   Notice that there are several default Signals associated with the channel.
3. Click **Add** to add another signal to **Digitizer Channels** > **Analog IN #0**.
4. In the **Add Signal** > **New name** field, enter **VmCmd**.
5. In **Signal Units**, select the unit prefix **m**, and enter the unit type as **V**.
6. For the **Scale Factor** value, click **Scale Factor Assistant**.
7. In the Scale Factor Assistant, select **Axopatch 200 series**, which indicate the amplifier you connected to Analog IN #0, then click **Next**.
8. In **Mode Setting** , select **V-Clamp** for voltage-clamp mode.
9. In **Ext. Command Input** , select **20 mV/V**.
10. Click **Finish**.
    *20 mV/V* computes and displays in the Scale Factor field.

## Setting the Holding Level

The holding level can be set in the following locations in the Clampex Software:

- On the Acquire >Edit Protocol > Outputs tab, by default holding levels for each output channel is set in the currently loaded protocol.
- In the Configure > Overrides dialog, select **Use Analog Holding Levels From Lab Bench**.
  One holding level is maintained on each channel irrespective of the protocol that is loaded. However, for wherever the override is set, immediate changes to the holding level can be made from the Real Time Controls.
- In the Tools > Membrane Test window, you can change the Hold level while running the test.
  The Membrane test returns the holding level to where it was at the time the test was opened.

## Telegraphs

A feature of the Clampex Software 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 200B amplifier, the state of the gain (α), filter frequency and capacitance compensation settings can be used by the Clampex Software.

Recall that you set the Gain of the Analog IN #0 signal "Im" to a value of 1, or unity gain. When using telegraphs, the Clampex Software 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.

To configure the telegraphs:

1. Select **Configure** > **Telegraphed Instrument**.
2. For **Telegraphed Instrument**, select the **Axopatch 200B** headstage is used.
3. In **Input Channels**, select **Analog IN #0** is connected to the scaled output of the amplifier.
4. Since you want to telegraph all three settings, connect the outputs of the Axopatch 200B amplifier to the telegraph ports on the Digidata 1550B digitizer.
5. Now, when you close the Telegraphed Instrument dialog and return to the Lab Bench, you see that the Telegraphs section at the bottom of the inputs tab reports the current status of the three telegraphed parameters.

**Tip:** If you enable the Telegraphs, you can see in the Scale Factor Assistant that the Gain setting is inserted via the Telegraphs. Any changes you make to these parameters are updated and used by the Clampex Software.

## Using Membrane Test

The first step in establishing a whole-cell recording is achieving a gigaohm 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). You will change the input signal to "Im", and the Membrane test output signal to "VmCmd" to match the signals that you created in the Lab Bench. You can also adjust the amplitude of the pulse when Membrane test is started and set initial holding levels. You 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 see the current response to square voltage steps.

You see the Clampex Software continuously report the Seal Resistance value. This value can be logged to the Lab Book at any time by clicking **=> Lab Book**.

To adjust how often the pulses are delivered, use the frequency slider.

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

## Editing Protocols

The details of an acquisition are defined in the acquisition protocols. As this is a fairly complex issue, this discusses only the features that are relevant to set up the example. In this case, this describes how to build two protocols: one that performs an I-V, and one that repetitively steps the cell to the same voltage. The software provides the 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\pCLAMP11.x\Sample Params.

## Protocol 1: Conventional I-V

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

### Mode and Rate

To set the Mode/Rate part of the example protocol:

1. Select **Acquisition Mode** > **Episodic Stimulation**.
2. Set the **Sweeps/Run** to a value of **15**, because the I-V will step from -80 to +60 in 10 mV increments, which needs 15 steps.
3. Keep the **Start-to-Start Intervals** time for **Sweeps** set to **Minimum** to have the protocol execute as fast as possible.
4. Since this will sample the data at 10 kHz, set the **Sweep Durations** 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. For this protocol example, 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. For this protocol example, set the **Channel #0** to the signal **VmCmd** that you created in the Lab Bench. This defines a **Holding Level** of **-80 mV**.

### Waveform

Defining the stimulus waveform on the Waveform tab is the next logical step.

1. For this protocol example, set the source of the **Analog Waveform** to be defined by **Epochs**, which are described by the table in the dialog.
2. **Digital Outputs** should be unavailable.
3. Leave the **Intersweep Holding Level** default value of **Use Holding** setting.
4. Use the **Epoch Description** table to define the actual epochs.
5. Define the **A**, **B**, and **C** epochs as step epochs by clicking in the **Type** table cell and selecting **Step** for each of the three epochs. For this protocol example, set the following Epoch Descriptions in the table dialog:
   You can use the <PgUp> and <PgDn> buttons to cycle through the epoch types.

**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, preview the waveform by clicking **Update Preview** at the bottom of the dialog. It is useful to have the Waveform Preview dialog open while you create the protocol, and click **Update Preview** every time you want to see the effects of something you have changed. It is handy to shrink the size of the Waveform Preview dialog and keep it displayed in a corner of the screen.

### Trigger

Leave the Trigger tab in its default state for this protocol.

### Stimulus

The Stimulus tab allows you to define additional stimulation parameters, such as presweep trains, leak subtraction and arbitrary values from a user list. For this demonstration, using the demo digitizer, you have not added P/N Leak Subtraction to the protocol on the Stimulus tab, however, when recording from real cells you might add P/N Leak Subtraction to the protocol. There is a P/N Leak Subtraction stimulus in the protocol, but it is left disabled. Select the checkbox to view the 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 Im. You can display the raw or corrected data.

### Statistics

The Clampex Software can display online various spike-oriented shape statistics on an acquired signal. Go to the Statistics tab of the protocol editor and select the **Shape Statistics** checkbox. In this example, you will measure the peak amplitude and time of peak on the “Im” signal.

If you were recording actual inward calcium currents, you would want to set the polarity of the peak detection to negative. Since you are using the demo driver, you will set the polarity to positive to illustrate the functionality of the statistics display. You can define a search region and a baseline region either in terms of time or epochs. You will 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 Region and first Search Region used in the peak detection.

The statistics can be saved whenever an acquisition is saved to disk, although if this option is not enabled, you can manually save the statistics from the Online Statistics window with **File** > **Save As**. You 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 you want to monitor the effect of a drug over time.

To create this protocol, you will modify the existing protocol that you 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 the protocol has been made 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. Click 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 the Clampex Software 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. You have learned about the Membrane test, which aids in the formation and monitoring of a gigaohm seal.

Before you save any data, define the file names and locations where the data will be saved. This is done in the File > Set Data File Names dialog. For this example, use “tutor” as the filename prefix. Select long filenames, disable the date prefix, and enter “tutor” in the file name field.

## Using Sequencing Keys

Sequencing keys are a powerful and flexible feature of the Clampex Software 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 you to set up a chain of events that define an entire experiment.

This section discusses 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 helpful to determine the flow of the experiment. In this 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.

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

**Table 5-2: Experimental steps, events, and protocols.**

| Step | Event | Protocol |
|---|---|---|
| 1 | Obtaining a whole-cell recording | |
| 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 you have established what the events are that you would like to define in the sequencing key setup, you 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:

- After you open the Sequencing Keys dialog, click **New Set** to define a new set of sequencing keys.
- To define a new event in the sequence, click **Add**. You are prompted as to what key you would like to define. Choose the default **Ctrl+1** for the **Add a Key Definition For** field.
- You can define the following operations that can occur when the key is activated: **Set Lab Bench Holding Levels** for Analog OUT channels, **Digital OUT Bit Pattern**, **Membrane Test**, **Protocol** and **Prompt** messages.
- For each key you define, use the Sequencing tab to define what happens when the operation completes. In this case, you link each event to the **Next Key**, so that you can set up a continuous sequence of events.
- You can define when the next event will occur. For example, you can specify that you would like the next event to occur **When Acquisition Finishes**, or **After (s)** a fixed number of seconds from the start of the key.

In the following example, you will define <**Ctrl+1**> to run the Membrane Test for 1 s, and when it completes, you 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 **Start Of Key**.
4. Click **OK** to finish defining <Ctrl+1>.
5. Click **Add** to add the <Ctrl+2> key.
6. Select **Operation > Protocol**.
7. Select **Action > Record**.
8. Click **Browse** and select the protocol file *Tutor_1*.
9. Set **Repetition Count** to **1** so that the protocol only runs once.
10. On the Sequencing tab, set **Next key > Ctrl+3 > When Acquisition Finishes**.
11. Click **Add** to add the Ctrl+3> key.
12. Select **Operation > Protocol > Action > Record**.
13. Click **Browse** and select the protocol file *Tutor_2*.
14. On the Sequencing tab, set **Next key > Ctrl+4 > When Acquisition Finishes**.

To toggle an electronic solution changer on and off, you can use the **Digital OUT Bit Pattern** on the Operations tab. Select the digital bit 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 this experiment, the **<Ctrl+4>** key is defined to flip Digital OUT #0 high:

1. Add the <**Ctrl+4**> sequencing key.
2. On the Operations tab, select **Parameters** and enable **Digital OUT Bit Pattern**.
3. Clear all bits except bit **0**.

   **Note:** You might have one bit unavailable here, overridden by the **Lab Bench > Output Signal > Set Digital OUT Bit High During Acquisition** checkbox.

4. On the Sequencing tab, set **Next Key > Ctrl+5**.

In this example, you want to keep the Digital OUT high until you set it back to low after you observe an effect of the drug on the cell. To keep the digital bits high when you run subsequent protocols, you must set in **Configure > Overrides > Use Digital Holding Pattern From Lab Bench**. When this option is active, any change to the digital outputs 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. Click **Copy** to create new keys, followed by **Properties** to edit the new key. When you are done the sequence should look like the following table.

**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 |

Save the sequence with **Save Set**, naming it something like “Tutorial.sks”.

After you close the Sequencing Keys dialog, you can start the experiment by clicking <**Ctrl+1**>, or the second icon button in the Sequencing toolbar.

A 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 acquire data, the Clampex Software provides some analysis functionality. It should be noted that data is best analyzed in the pCLAMP software, but the Clampex Software allows you to browse data and perform simple measurements.

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

When the data opens in the Analysis window, there are four cursors that can be moved by clicking and dragging. Many of the options are available by clicking the right-mouse button.

For now, you will measure the peaks of the calcium currents that you acquired during the last I-V of the sequencing series, which was the last file that was recorded, and which should now display. You will measure the minimum value between cursors 1 and 2:

1. Set cursors 1 and 2 so that they delimit the region of the peak inward calcium current.
2. 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.
3. Double-click on the Results window at the bottom of the screen, or select **Window > Results**. Observe that for each trace, a number of parameters were measured.
4. Select **View > Window Properties** when the Results window displays to specify which measurements to display and in which order they display.
5. 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.

### Real-Time I-V Plotting in pCLAMP software

Despite the power of the Clampex Software, you will now coordinate its acquisition with I-V plotting in the pCLAMP software. For this demonstration you may want to return the pCLAMP software 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 **Molecular Devices > pCLAMP 11** program folder. Select **Clampfit** as the application to return to default values.

Now load the pCLAMP software and the Clampex Software and in the Clampex Software load the protocol file *tutor_1* or the protocol file provided as *tutor_1a.pro*. In the pCLAMP software, check **Apply Automatic Analysis** in **Configure > Automatic Analysis**. Use the Record function in the Clampex Software (do not use the sequencing keys yet) to record a file, which then displays in an Analysis window in the pCLAMP software.

You will now use this file to create the I-V plotting routine and then you will set up the pCLAMP software to run an Automatic Quick I-V Graph concurrently with the experiment in the Clampex Software:

1. With the focus on the Analysis window in the pCLAMP software, select **Analyze > Quick Graph > IV**. You want to plot the peak current Im against the command voltage VmCmd.
2. For the **X Axis Waveform** group, select **Epoch**, and for the X axis of the plot, select **Epoch B level**.
3. For the Y Axis, for Signal choose **Im**. You want to detect the positive peak in Epoch B so select Peak and select **Positive** for the Polarity. For Region select **Epoch B - Waveform 0**.
4. You apply a smoothing factor of **5** points in peak detection.
5. Select **Destination Option > Append** because you want to see all I-V curves in the experiment plotted together.
6. When you close the form a Graph window displays an I-V plot.
7. Minimize all applications on the computer desktop except the Clampex Software and the pCLAMP software. Move the mouse to a blank part of the Windows taskbar, right-click and select **Tile Windows Horizontally** to give equal place to the two applications. You may want to arrange the Windows within each application to make maximal use of the desktop space. In the pCLAMP software minimize or close all windows except the Analysis window and the Graph Window and arrange the windows with the **Window > Tile Vertical** command in the pCLAMP software.

8. Before you begin the experiment, open **Configure > Automatic Analysis**, check **Generate Quick Graph** and select **I-V**. In this demonstration, make sure that **Prompt for Arithmetic** is clear. 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 the Clampex Software, 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.

## Displaying and Measuring Acquired Data

After you acquire data, the Clampex Software provides some analysis functionality. It should be noted that data is best analyzed in the pCLAMP software, but the Clampex Software allows you to browse data and perform simple measurements.

You can open data files by selecting **File > Open Data**. This dialog provides options that control how the data displays once it is read into the Clampex Software. 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 to display. Open the last data file that was recorded by selecting **File > Last Recording**.

When the data opens 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.

For now, you will measure the peaks of the calcium currents that you acquired during the last I-V of the sequencing series, which was the last file that was recorded, and which should now display. You will measure the minimum value between cursors 1 and 2:

1. Set cursors 1 and 2 so that they delimit the region of the peak inward calcium current.
2. Select **Tools > Cursors > Write Cursors** or click the **Check** toolbutton at the top left of the Analysis window or from the Analysis toolbar.
3. Double-click on the Results window at the bottom of the screen, or select **Window > Results**. Observe that for each trace, a number of parameters were measured.
4. Select **View > Window Properties** when the Results window displays to specify which measurements to display and in which order they display.
5. Save these measurements to a text file by selecting **File > Save As**.

There is extensive control over various features of the Analysis window. The Help file covers this in detail.

## Real-Time I-V Plotting in pCLAMP software

Despite the power of the Clampex Software itself, you will now coordinate its acquisition with I-V plotting in the pCLAMP software. For this demonstration you may want to return the pCLAMP software 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 Molecular Devices > 11 folder. Select **pCLAMP software** as the application to return to default values.

Now load both the pCLAMP software and the Clampex Software and in the Clampex Software load the protocol file *tutor_1* or the protocol file provided as *tutor_1a.pro*. In the pCLAMP software, check **Apply Automatic Analysis** in **Configure > Automatic Analysis**. Using the Record function in the Clampex Software (do not use the sequencing keys yet) record a file, which then displays in an Analysis window in the pCLAMP software.

You will now use this file to create our I-V plotting routine and then you will set up the pCLAMP software to run an Automatic Quick I-V Graph concurrently with the experiment in the Clampex Software:

1. With the focus on the **Analysis** window in the pCLAMP software, select **Analyze > Quick Graph > IV**. You want to plot the peak current Im against the command voltage VmCmd.
2. For the **X Axis Waveform** group select **Epoch**, and **Epoch B Level** for the X axis of the plot.
3. For the Y Axis, for **Signal** select **Im**. You want to detect the positive peak in Epoch B so select Peak and select **Positive** for the Polarity. For Region select **Epoch B - Waveform 0**.
4. You apply a smoothing factor of 5 points in peak detection.
5. Under **Destination Option** select **Append** because you want to see all I-V curves in the experiment plotted together.
6. When you close the form a Graph window displays an I-V plot.
7. Minimize all applications on the computer desktop except the Clampex Software and the pCLAMP software. Move the mouse to a blank part of the Windows taskbar, right-click and select **Tile Windows Horizontally** to give equal place to the two applications. You may want to arrange the Windows within each application to make maximal use of the desktop space. In the pCLAMP software minimize or close all windows except the Analysis window and the Graph window and arrange the windows with a **Window > Tile Vertical** command in the pCLAMP software.
8. Before you begin the experiment, open **Configure > Automatic Analysis**, activate **Generate Quick Graph** and select **I-V**. In this demonstration, make sure that **Prompt for Arithmetic** is clear. 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 the Clampex Software, 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**. Start the experiment by clicking 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 provides an overview of the performance features of the Membrane test. It is important that you proceed through the tutorial in sequence.

The experiment consists of:

1. To set the Clampex Software to demo mode. Go to the Configure > Digitizer dialog, click **Change** and select **Demo** from the **Digitizer Type** list.
2. Open the **Membrane Test**. Click **Circuit Diagram** in the toolbar or select **Tools > Membrane Test**. Confirm the OUTPUT signal is Cmd 0 (mV) and that the INPUT signal is I_Mtest 0 (pA). If not, change the settings in **Configure > Membrane Test Setup**.
3. Select the **Membrane Test** mode. Click **Cell** and confirm the input channel **#0** is selected. Click **Play** to start the Membrane test.
4. Change the voltage 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. Notice that the capacitive transients from the current record in the Scope display decrease in amplitude. Change the Pulse amplitude back to **20** mV and observe that the capacitive currents are larger.
5. Click the up arrow button on the left side of the window by the Y axis to change the amplitude scaling. Click **Auto Scale** next to the meters and observe the change.
6. Move the **Slower – Faster** slider to change the measurement update rate. 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. Select the **Averaging** checkbox to enable averaging. Notice that the noise of the capacitance signal decreases in the Online Statistics window. Click the **Averaging** checkbox again and observe the noise increase (wait a moment for the new data to appear). Notice that the recording speed displayed below the **Slower – Faster** slider increases dramatically.
8. Open **Configure > Membrane Test Setup** to change the number of pulses in the pulse train. In the Pulse Train Protocol section, set **Number of Pulses In Train** to **4**. Change the step level (mV) to **40**. Click **OK** to return to the Membrane Test window. Click **Pulse Train** 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. Click **Lab Book** and 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 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 the pre-tutorial settings in the Membrane Test Setup dialog.

## Scenarios

### Cell-Attached Patch-Clamp Single-Channel Recording

Use the Clampex Software Membrane Test tool 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 store 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 high, Gap-Free recording is 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 load one protocol or the other, based on your initial impressions of the data. Your subjective experience of your preparation and your experimental goals 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 an experimental measurement.

For example, if you 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 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 factor, and 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 to make decisions about how to best get the 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 you 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). 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. On the **Trigger** tab of the Edit Protocol dialog, you can select **Options > External Tag**, and use the **TRIGGER IN/TAG BNC** on the digitizer to mark changes in the file while recording.

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. If you want to see if there are immediate (for example non-stationary) effects of the voltage jump, then you can change the $V_{out}$ signal in real time and have a comment 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, 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 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 be a problem with threshold-based event detection. Event-triggered acquisition can 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 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 Software Lab Bench are applied to each channel separately, this can be accomplished within the Clampex Software. For example, the Clampex Software 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 a function of the filter setting on the trigger channel and a function of the filter setting on the data channel. Single, brief events may not trigger acquisition. Brief events within bursts of activity resolve at the higher bandwidth. Therefore, this approach should be applied to the analysis of kinetics within bursts of activity. If you change the holding potential while conducting a variable-length event-driven acquisition, you should also keep in mind that the change in channel amplitude alters the event-detection sensitivity.

## Isolated Patch-Clamp Single-Channel Recording

The big advantages of isolated patch-clamp recording over cell-attached recording is that it provides 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). 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.

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. 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 the Clampex Software. 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 Trigger Source > Digitizer START Input or to an acquisition channel. If a signal from the DAS is sent to the Digitizer START Input, then the Digitizer START Input must be appropriately configured in the Edit Protocol dialog on the Trigger tab. Once a start signal is received, acquisition continues until you click **Stop**, 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 clicking **Record**. Data are not written to disk until a signal is received from the DAS. Acquisition can be terminated by clicking **Stop**, a preset duration, disk space limitation, or by the return of the DAS signal to a value below the trigger level.

A DAS may 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 in the Clampex Software may be used and configured to tag the changes made to the output channel that provides the voltage-step commands to the DAS. 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. 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 that contains periods of channel activity that corresponds to agonist (or other drug) applications can be prepared for analysis by opening the file into a Clampex Software Analysis window and using cursors 1 and 2 to delineate segments of interest. **File > Save As** can be used to save each segment as a .abf file for subsequent analysis in the pCLAMP software.

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 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 captures any class of events that transpire within an anticipated length of time. This mode creates a sequence of sweeps that can be averaged or analyzed for individual shape statistics in the pCLAMP software.

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 most practical to coordinate the DAS and the acquisition with an external pulse generator. The pulse generator sends a signal in parallel to the DAS and the digitizer's Analog IN channel to serve as the event trigger input. However, it is possible to use an Analog OUT signal from the digitizer for the same purpose. Using Real Time Controls in the Clampex Software, 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 Edit Protocol dialog, 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 **Record**. When you want to activate the valve controller, you can enter **5** in the text field for Analog OUT 0 and then trigger the valve when you click **Enter**. This should generate a response and a record of the relative timing of the valve control signal. To turn off the valve, enter **0** into the text field and click **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 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 the pCLAMP software, instead of having to first rerun Event Detection on the raw data in the pCLAMP software to recapture the events for subsequent analysis.

## Whole-Cell Recording of Evoked Activity

In most cases, Fixed-Length Event-Triggered acquisition is best to record 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 acquire is in the form of spike trains rather than single responses. 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 record 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. Double-click on the Scope window to display 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 entering the level in the Real Time Controls text field, and then waiting for a specific time mark (for example an elapsed time value) before pressing the Enter key. The Analog OUT only changes once the Enter key is pressed. This way, a complete experiment containing multiple drug applications can be saved as a single file.

To record a complete experiment in a single file but omit long times of inactivity such as when the drug is being washed out, the Clampex Software can be toggled between the Record and View Only modes using the **Pause-View** 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.

# Chapter 6: Clampfit Analysis Software Features


This chapter is an overview of the Clampfit Software that outline the main features and organization of the program. Of special interest are the sections on event detection and single-channel analysis in the pCLAMP software. Included in the single-channel analysis section are guides for Fetchan and pSTAT.

## Clampfit Software Windows

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

As in the Clampex Software, all commands are included in the main menus, though many of these also have toolbuttons or can be accessed from right-click menus. Different menus are available according to the window type that has the focus, for example 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 Software 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 menus with options specific to the window, including in each case one that opens a Window Properties dialog. This allows you to configure the general appearance of the window, and these settings can then be saved as defaults in **View** > **Window Defaults**.

The basic features of each of the window types are set out in the following, but also see the Help file 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 displays graphically, as traces, with sub-windows 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 compared with the original (**Edit > Create Duplicate Signal**). 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 in Analysis windows from **File > Open Data**, or configure the Clampex Software and the pCLAMP software to open newly recorded files in the pCLAMP software (**Configure > Automatic Analysis**). Configure the way files open into a new Analysis window or into an existing one to replace the previous file, in **File > Open Data** Options.

Data can display as Sweeps, or in Continuous or Concatenated modes, controlled from **View > Data Display**. In sweep mode you can select to view a subset of the sweeps (**View > Select Sweeps**), and toggle between viewing all 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 50 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 on 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 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 display the Cursor Properties dialog.

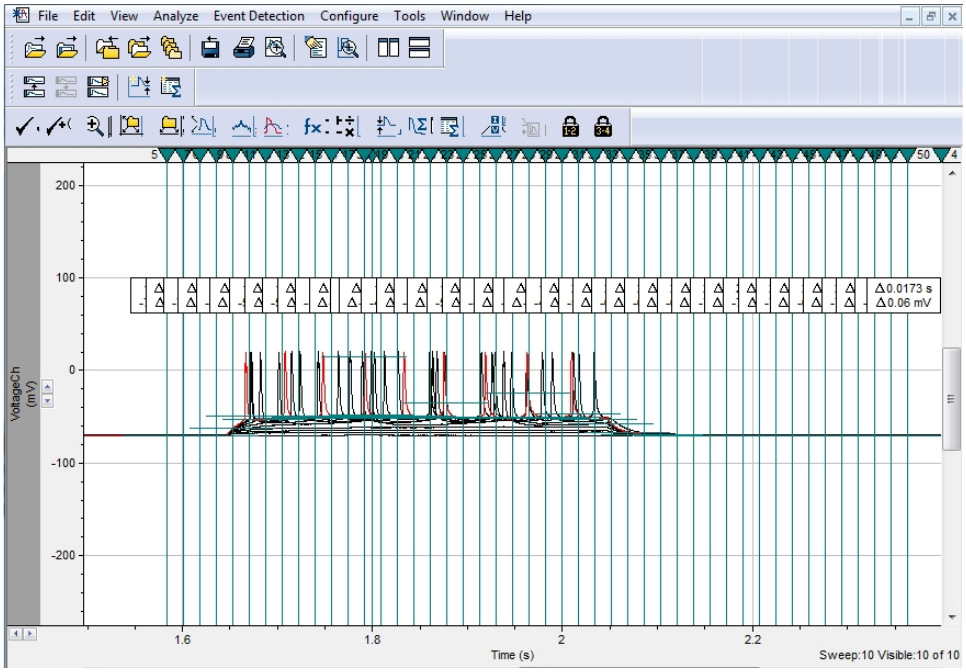



**Figure 6-1: Set Up to 50 Cursors**

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 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 the parameters reported for each file. These options give you great flexibility in being able to find and organize files from particular types of experiments. This is 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 data in two-dimensional graphs. A range of graph types is available, such as scatter and line plots, and histograms. Any graph can contain multiple plots, in which case a legend displays on the right-hand side of the window. Graphs can be named and the axes labeled in the View > Window Properties dialog, which contains other configuration options. Axes for most graph types can be toggled between linear and logarithmic scaling with **View > Axis Type**.

Data manipulation options and analyses are available to apply to Graph window plots, including curve fitting, normalization and square root, amongst others 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 be created from any Results window data and when a graph is generated from some other source (for example a data file in an Analysis window with **Analyze > Quick Graph > Trace vs. Trace**), the data that displays in the graph is also written to a sheet in the Results window (for a Quick Graph, to the Quick Graph sheet). To save a graph, you must save the Results window that was used to generate the graph. If you manipulate data in the Graph window, the corresponding data in the Results window changes to the new values you set in the Graph window.

Graphs are generated in the pCLAMP software in a number of ways:

- From an Analysis window, the **Analyze > Quick Graph** command has options to generate I-V and Trace vs. Trace graphs. These graphs can be set to update automatically for data files received from the Clampex Software, in **Configure > Automatic Analysis**. They can 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 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 from a sheet in the window by selecting data in a column and using the **Analyze > Create Graph** command. The 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. The Analyze > Assign Plots dialog provides more options to create graphs.
- Several of the analyses in the Analyze menu (for the Results and Analysis windows) generate a graph as part of their output, for example the Kolmogorov-Smirnov Test, Autocorrelation and Nonstationary Fluctuation Analysis. In each case, the data in the graph is 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 for use with events data, which is found on the Events sheet. It can 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 that corresponds to it in a Define Graphs scatter plot. See Event Detection on page 102.

## Lab Book

The Lab Book window is a text editor to log events that occur while the pCLAMP software runs. You can set how much you want to write to the Lab Book with **Configure > Lab Book Options**, and add comments with **Tools > Comment to Lab Book**, or in a editor fashion in the window. The results of several pCLAMP software analyses, for example, 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 for editing or archiving elsewhere.

## Layout Window

The Layout window is a page layout editor to format spreadsheets and graphs 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 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.

## Results Window

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

Many analyses, including fitting, are available for data presented in the Results window, and all graphs have the data stored in Results window sheets. The results of many analyses are recorded in the Results window, often on sheets dedicated to results from a particular analysis. New results may either be appended to previous values or written over previous data. If data you try to append data that 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 and Results |
| Correlation | Analyze > Autocorrelation and Cross-Correlation | Analysis, Graph and Results |
| Fluctuation | Analyze > Nonstationary Fluctuation | Analysis |
| Histogram | Analyze > Histogram | Analysis, Graph and Results |
| Power | Analyze > Power Spectrum | Analysis |
| Resistance | Analyze > Resistance | Analysis |
| V-M Analysis | Analyze > V-M Analysis | Analysis |
| Quick Graph | Analyze > Quick Graph | Analysis |

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 computer clipboard. Data can be copied into the open sheets from the Analysis window by using the **Edit > Transfer Traces** command. Statistics (.sta) files opened in the pCLAMP software transfer the data into a numbered Results window sheet and open in a Statistics window.

Only one Results file can be open at a 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 the graphs associated with the window at the same time.

## Statistics Window

Statistics files created in the Clampex Software can be opened in the pCLAMP software from the File > Open Other > Results & Statistics dialog. As well as opening into a Clampex Software-Style Statistics window, the data transfers to the first empty unreserved Results window sheet.

## File Import

The Windows environment in which the Clampfit Software operates permits easy transfer of data and figures between the Clampfit Software and other applications in the same environment. Simple cut and paste operations allow you to bring analytical results from other programs into the Results sheet. You can open any text file into a Results sheet. Open the File > Open Other > Results dialog and select the text file to open. A dialog opens for you to select the column delimiter in the file and set other options, before the file displays.

The Clampfit Software imports numerical files for graphical display in the Analysis window. Any simple ASCII or binary output file can open in this window. When a non-Axon Instruments text or binary file is selected from **File** > **Open Data**, a conversions dialog displays. You need to convert the file into the scaled data that display, specifying the number of signals present in the file and assigning signal names, units and scaling factors. A preview of the source file allows you to specify a number of lines to skip 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 the Clampfit Software generates the time values when you import the data.

## Data Conditioning

After raw data is read into the Analysis window, the Clampfit Software 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 Digital Filters on page 183.

Files can be baseline-corrected and spurious activity edited out. **Analyze** > **Adjust** > **Baseline** applies baseline adjustment as a fixed offset, or as an offset of the mean of the entire file, or of a particular region of the file. If there is baseline drift in a sweeps file, for example, subtracting the mean of the first holding period can bring each sweep to a common zero-value baseline suitable for further analysis.

The Clampfit Software can 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 allows you to define the baseline in data which requires complex correction functions. When you select this option, a line is drawn over the data. You define the 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 before export. See Figure 6-2.

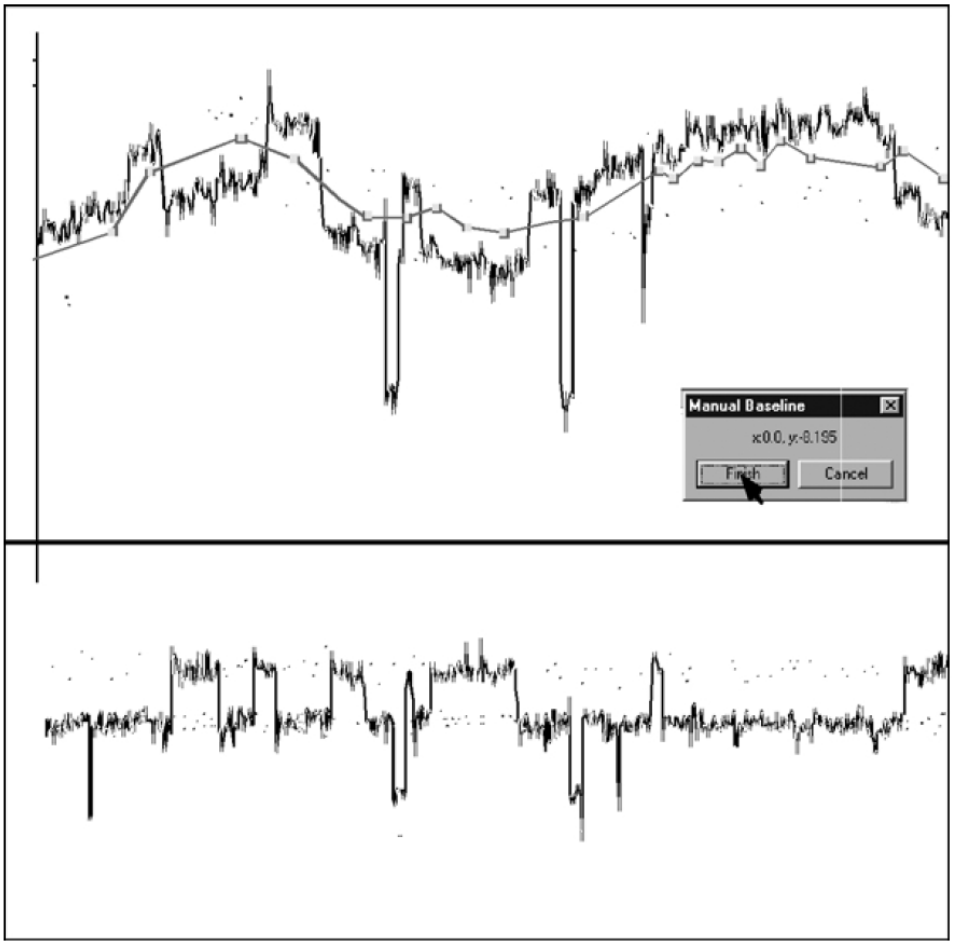



**Figure 6-2: Manual baseline adjustment.**

Another way to prepare 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 multiply the resultant file by the 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 remove unwanted artifacts from files with Analyze > Force Values, or save sections of data files by using 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 (for example, 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 an integrated mode of operation that binds the functionality of graphs, the Results window, and other features for event detection with the Analysis window where the file displays. One data file at a time can be searched for trace features that mark a biological event, such as a synaptic potential or a single ion channel opening. The main types of event detection search include:

- **Template Search** - 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 Search** - Amplitude baseline and threshold levels are set, then the file searched for data that crosses the thresholds.
- **Minis Search** - Minis search is used to analyze minis which includes miniature postsynaptic currents (mPSCs) or potentials (mPSPs). It provides a streamlined workflow for event detection. It enables you to add, delete, or edit events in a simplified way. You are able to adjust the baseline or duration of events for accurate measurements. Saving Minis search file is available.
- **Single-Channel Search** - Single-channel records are translated into idealized events, which are categorized as belonging to the levels that correspond to different ion channel amplitudes in the patch.
- **Population Spike Search** - A system to detect and measure population spikes of various configurations including single and multi-peak responses and paired field responses.
- **Action Potential Search** - Action Potential searches for measurements and enables you to adjust baselines and select measurements such as amplitude, action potential, action potential duration, peak to peak frequency and time, and threshold potential.

### Event Detection Searches

Configure searches with conditions to filter out false hits and to define events to ensure that event statistics taken for each event found-measure parameters in the way you expect. Threshold and template searches can have up to nine categories defined and concurrently searched for when you run a search.

All searches can accommodate high degrees of baseline drift provided this does not include abrupt shifts. You can change levels while a search session is in progress by dragging level markers in the Analysis window to find events in accord with the new levels. Statistics for the events are recorded relative to the new levels. Levels are the only search parameters that you can change on the fly. To change any other parameter you must stop the search, reconfigure the dialog and then click **OK** to restart the search.

When a set of event search parameters is confirmed (by clicking **OK** in the Search dialog) an event detection toolbar is inserted into the Analysis window that contain the file being searched. Use this to control the progress of the search. The first candidate event is highlighted to 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 software 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. Once you conclude this, the next candidate event is found and the process repeats. You can 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 found events, and sit back and watch these accumulate. Once a search finishes 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 can view events data, as they accumulate, in five other locations: the Event Monitor, Event Viewer, Graphs window, 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.

## Event Monitor

Event Detection > Event Monitor reports key statistics measurements for the current candidate event to provide a comparison with previously accepted events for that category (for peak-time events) or level (single-channel events). Use this when you assess 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. Measurements are 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 are defined and confirmed. Each graph can be configured as a conventional or logarithmic histogram or scatter plot, with the measurements recorded in the Results window available to plot. These graphs dynamically update as new events are found.

## Event Viewer

The Event Viewer (**Event Detection** > **Event Viewer**) contains an Analysis window that accepts the trace segments identified as events in the 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

A summary of the statistics measurements recorded so far in any session display 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 the 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), shows that event alone in the Event Viewer, and shows the values for the event in the Event Monitor.

# Single-Channel Analysis

The following topic are guide for Fetchan software and pSTAT software users upgrading to the pCLAMP software. They provide an outline of where functionality from the old single-channel analysis programs is found in the Clampfit Software. It is assumed that you are familiar with general navigation and data display commands within the pCLAMP software.

## Fetchan Software Compatibility

The following explains how to use data files from Fetchan software in the pCLAMP software.

### File Subtraction

Use **Analyze** > **Subtract Control** to subtract a control file from a test file. Use this dialog to scale the control file before it is subtracted. Subtraction can be set up to be applied automatically as files are imported into the Clampfit Software, with **Configure** > **Automatic Analysis** > **Subtract Control** file.

Any further baseline correction must be done as a separate operation. Use **Analyze** > **Adjust** > **Baseline**, where there are a number of options for this.

The Clampfit Software has no direct equivalent to the Fetchan Software subtraction option, **Closest N Episodes**. To get an equivalent result, subtract an averaged trace and then apply baseline adjustments.

For example, to bring a set of sweeps with rising or falling baselines to a common zero baseline, subtract the mean of a trace segment that has no activity in any of the sweeps. Typically the first holding fits this criterion.

### Data Modification

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

- **Filter Freq** - Filtering is performed in the Clampfit Software in **Analyze** > **Filter**, where you have filtering options. Unlike Fetchan Software, you must filter a file as a separate step before you run 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 you 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 the Clampfit Software, are all recorded in the file header. If you record without telegraphing the amplifier filter frequency, you should set the Clampex Software **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 Single-Channel Event Amplitudes on page 145.
- **Derivative** - convert a data trace to a plot of its differential values with the diff( ) function in **Analyze** > **Arithmetic**.
- **Change Polarity** - change the file scale factor.
  To do this:
  - Open **Edit** > **Modify Signal Parameters** and reverse the polarity of the scale factor.

    **Note:** This can only be done with .abf files that have not been modified in any way within the pCLAMP software.

### Analysis Mode

In Fetchan Software, different kinds of analysis are available by changing the operation mode in the **Parameters** > **General** > **Analysis** > **Analysis** mode field, which then shows the Analysis menu commands relevant to the analysis mode. The Clampfit Software provides available analyses after you open a file in an Analysis window. The following Clampfit Software functions replace the Fetchan Software modes.

### Episode Average

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

To select 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 go 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 a .abf file (**File** > **Save As**). In the Save As dialog, use the **Data Selection** option, **Visible Sweeps And Signals** so that the saved file contains only the averaged sweep. This is because all the original sweeps are still in the file, simply hidden from view. If you use a file that contains these as a control file for subtraction, all the hidden sweeps and the averaged sweep are used in the subtraction.

### Segmented Average

The functionality in this operation mode is found in the Analyze > Segmented Average dialog in the Clampfit Software. The segmented average trace is built up in the Clampfit Software in a similar way to Fetchan Software, with you 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 the Clampfit Software, the entire sweep is offered for creation of the segmented average, and the resulting trace saved from the dialog is for the entire sweep.

### Pulse Average

The Clampfit Software has no dedicated command to generate pulse averages, however this Fetchan Software function can be 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, click **Copy As** in the Viewer to save the assembled events as a .abf file. Then average the events with **Analyze**> **Average Traces** to create the pulse average trace.

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

### Events List

This Fetchan Software operation mode identifies level transitions in Gap-Free and Variable-Length files. **Event Detection** > **Single-Channel Search** replaces it in the Clampfit Software. Resulting data writes to the Results window Events sheet showing similar information to Fetchan Software EVL files.

In Single-Channel searches an idealized record is superimposed over the trace being searched, to display 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 Software, set in **Parameters** > **Special** > **Events List Analysis**, is carried out within the Search Configuration dialog in the Clampfit Software. 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:** 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 the threshold (50% crossing) for categorization of an event in either level. As in Fetchan Software, 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 in Fetchan Software was performed at the same time as the creation of the idealized event trace, must be carried out independently in the Clampfit Software, with **Analyze** > **Filter**.

To run the search, much Fetchan Software 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.

**Table 6-2: Fetchan Software and Clampfit Software Command Name Mapping**

| Fetchan Software | Clampfit Software |
|---|---|
| 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 given to an event you can adjust this by dragging the idealized event-defining lines in the Analysis window.

## Brief Events

In Fetchan Software, Short events are called Brief events in the pCLAMP software, and labelled B in the Events sheet, in the State column. They can be excluded from analyses you want do with the events data.

The pCLAMP software uses the same algorithm to determine brief events as Fetchan Software. 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 the amplitude cannot be calculated in the normal way. In these cases the pCLAMP software reports the amplitude of the midpoint of the event as the amplitude of the entire event. See Single-Channel Event Amplitudes on page 145.

## Latency

Fetchan Software 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 (for example the latency) is measured. In the pCLAMP software 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 search, 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 position above to define the search region.
3. Select the **Latency Analysis** checkbox in the Search Configuration dialog.

With the **Latency Analysis** checkbox selected, 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 and selection of the cursors for the search region ensure these values are measured from the time the stimulus was delivered. This allows the software 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.

You can set an Ignore Initial Duration setting in the search configuration dialog. This is equivalent to Fetchan Software Parameters > Special > First Latency Analysis Ignore setting, whereby level transitions for the stipulated period following the start of the search region are ignored. Use this 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 that contains the file is selected. 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.

## pSTAT Software Compatibility

pSTAT software takes as its usual input the events list (.evl) files generated in Fetchan software. Equivalent data in the Clampfit Software 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 software.

### Histograms

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

1. Select the data column to display in a histogram.
2. Select **Results** > **Histogram**.
3. From the dialog, select the histogram type and bin width, and ensure that you have the correct Results window sheet and Data > All Selected Columns.

   **Tip:** To include only a specific row and/or data range, restrict the data from the column.

4. Click **OK** to generate the histogram.

By selecting more than one data column, you can include more than one parameter in a histogram graph, with their values combined or displayed separately, color-coded. You can normalize the area under the histogram.

The histogram function does not differentiate between data according to level or any other parameters. To do this, you need to use the **Event Analysis** > **Fast Graph**.

### 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 dialog is designed for events data. It includes most of the options in the Histogram dialog and 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 the pCLAMP software 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.

The pSTAT software 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, click **Convert to Frequency** to convert the count for each bin into a frequency, found by dividing the count by the bin width.

### Burst Analysis

Burst Analysis in the Clampfit Software is carried out from Analyze > Event Analysis > Burst Analysis. The burst analysis in the Clampfit Software differs in a number of respects from that in pSTAT software. There are two way to find bursts in the Clampfit Software:

- 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 the Clampfit Software inter-event interval is the same as the inter-burst interval that would be used in pSTAT software, the difference being that the Clampfit Software looks for closed-channel times less than the interval in order to cluster events together as belonging to a single burst, while pSTAT software 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. See Burst Analysis on page 151.

### 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 provides the option to generate a P(open) histogram similar to those in pSTAT software. As in pSTAT software, you can manually enter an interval setting for this, or have it done automatically by the Clampfit Software. See also P(Open) on page 153.

### 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. 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 configure the Single-Channel search (Event Detection > Single-Channel Search) appropriately. This involves selecting **Latency Analysis**, and setting the event detection search region to start at exactly the time of the stimulus. See Fetchan Software Compatibility on page 104.

### Peri-Event Analysis

This analysis is new in the Clampfit Software, with no equivalent in pSTAT software. It measures the intervals between events and events that occurred within a stipulated time range. 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 entering **T** into the State column in the lines of the events on the Events sheet. See Peri-Event Analysis on page 152.

### Idealized Trace

The Clampfit Software 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. After the event detection session is closed, the idealized trace cannot be re-created.

## Curve Fitting for Single-Channel Results

The pCLAMP software provides the Gaussian function to fit to amplitude data and exponential probability functions to fit 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. The pCLAMP software provides an 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 6-3 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.

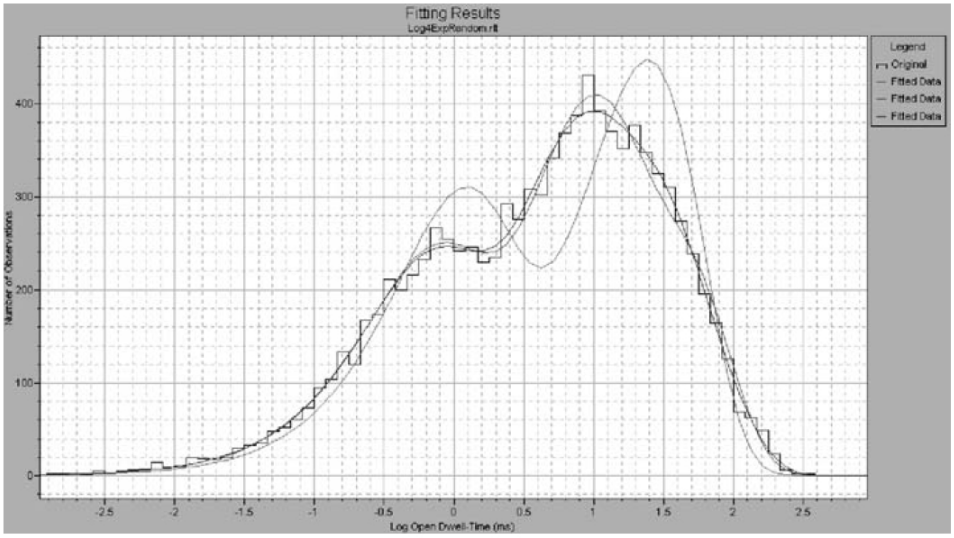



**Figure 6-3: Variable metric fits with 2, 3 and 4 exponential terms.**

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

The pCLAMP software provides an option to statistically determine the best fitting model (number of terms) for dwell-time or amplitude distributions; select **Compare Models** on the Analyze > Fit > Data/Options tab. Fitting models are iterated automatically. 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 display along with the test statistics so you can evaluate the results without referring to statistical tables. This information is provided following the fit on the Analyze > Fitting Results > Statistics tab.

For example, Figure 6-4 shows the results of an automatic model comparison for the log-binned data shown in Figure 6-3.

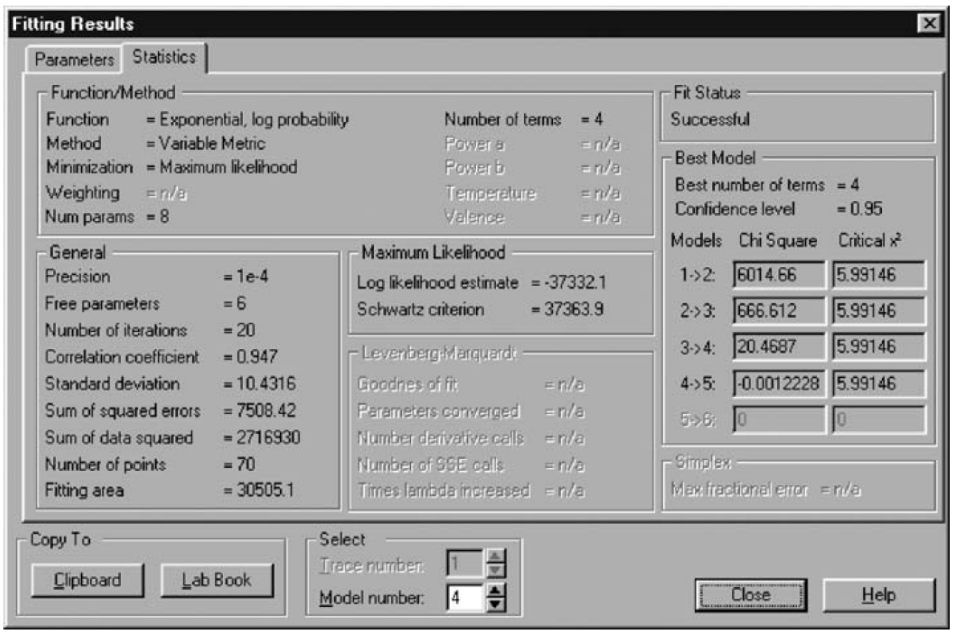



**Figure 6-4: Fitting Results dialog Statistics tab.**

The model comparison statistics display to the right in this dialog. 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

The pCLAMP software 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 pCLAMP software 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. 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. You should read the information and the references provided on the QuB web site before you use the QuB programs to analyze their data. The QuB methods are best suited for data that have not been heavily filtered.

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

To preprocess in the pCLAMP software, data to be analyzed by QuB should be baseline-adjusted before export, best accomplished using the manual adjustment method. 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 contains multiple kinds of channel activity, distinguishable for example, by amplitude, use the Blank command to create a sequence of homogeneous segments, suitable for fitting to a single model.

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 the pCLAMP software. Once the Xaxis data in the QuB duration histograms is converted to Log values, using **Analyze > Column Arithmetic** in the pCLAMP software, the SKM-idealized data can be fit with conventional methods. This approach permits the direct comparison of QuB and the pCLAMP software idealizations.

## Fitting and Statistical Analysis

The pCLAMP software boasts a wide range of predefined fitting functions for data that displays 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 display along with the data, fitting in these windows is only suitable for simple X-Y pairs of data. Data in a Results window can be fit with complex functions such as the Goldman-Hodgkin-Katz equation by assigning multiple variables to specific columns.

Fitting in the pCLAMP software is dealt with thoroughly in Curve Fitting on page 209 and Fitting Functions on page 241.

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 the epoch regions to which to apply the analysis from within the analysis configuration dialog. If you do not want to analyze the whole file, set a cursor pair about the region to analyze before you open the configuration dialog, then select this region after you open 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 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 an 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 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. The pCLAMP software is configured with 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. Use the Layout window 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 the pCLAMP software come together for final preparation and presentations. Graphs, raw data traces, and tabulated data can be copied 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 the pCLAMP software. 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 an application such as Word, and from there, transfer them to the target application.

# Chapter 7: Clampfit Analysis Software Tutorials


The scenarios presented in this chapter introduce the Clampfit Software analysis functionality by implementing procedures that are 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. After you master these features, you can analyze more directly.

The scenarios highlight some of the features discussed in the last chapter. The tutorials give menu commands for each step in the analysis, but you should become familiar with toolbuttons and hotkeys. Also try the context-sensitive commands in the right-click menus. Context-sensitive online help is available in the program when you press the <F1> key or the Help buttons.

The sample files are found in the ...\Program Files\Molecular Devices\pCLAMP11\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. To save the analyzed files you should not replace 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.

**Note:** The pCLAMP software can manipulate data in illegitimate as well as legitimate ways. It is your responsibility as a scientist to consider the changes you introduce into the data and to maintain the integrity of the results.

## Creating Quick Graphs

The pCLAMP software can generate current-voltage plots with just a few clicks. In cooperation with the Clampex Software, you can create I-V plots automatically, in real time, immediately after the application of a step protocol. See Real-Time I-V Plotting in pCLAMP software on page 87.

### 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 that shows outward currents in response to a voltage step protocol. To see the protocol that was applied, select **View** > **Stimulus Waveform Display**, or click the toolbutton in the main toolbar. From a holding potential of -50 mV, depolarizing voltage steps were applied using the Analog Output Channel AO #0. This tutorial 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. Drag the cursors to the new positions or double-click the cursor number, selecting **Time** and entering the X axis value in the **Time** field, then click **OK**.
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 (Voltage): specify it by the positions of cursors 1 or 3, or define it using epochs. In the example, place all four cursors 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 (Current)**, plot the **Peak** outward currents elicited during the voltage step. There is only one **Signal** > **IN 0 (A)**, which you can specify. To avoid the capacitive transient at the onset of the voltage step, restrict the search for a **Peak**> **Polarity > Positive** to the **Region**> **Cursors 1..2**.
5. The pCLAMP software allows you 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 must 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 you will learn how to compare a filtered signal to the original.
6. For comparatively slow currents, set **Smoothing Points** to **11** to provide a sufficiently smooth signal with little danger of over filtering.
7. Click **OK**.

The pCLAMP software creates a new Graph window with the I-V plot, which can be customized after double-clicking to display the Properties dialog, or by using the items in the View menu.

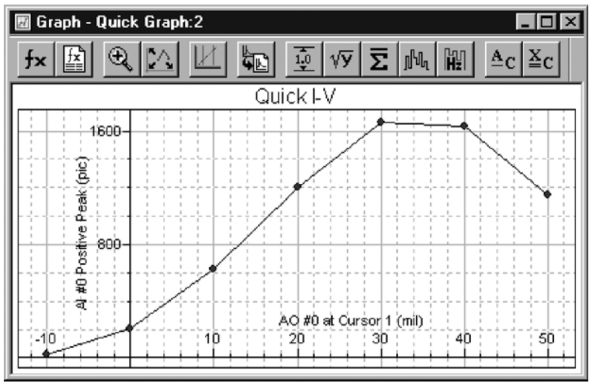



**Figure 7-1: Quick I-V graph for cav.dat.**

## Quick Trace vs. Trace Plots from Whole-Cell Currents

The application of voltage ramps frequently contributes to the characterization of ligandgated 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 allows plotting different elements of a data file versus each other sample by sample. The dialog provides two sections to define the X Axis and Y Axis values. The **Trace**, **Waveforms**, and **Signal** options are available from drop-down lists. You can Invert the values to account for data from cell-attached or inside-out recordings.

To plot the only available sweep of the input signal AI #15 versus the output signal AO #0, 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* you had to avoid the capacitive transients at the beginning of the voltage steps, you now perform the analysis on the entire ramp epoch. Therefore, specify **Region** > **Epoch C - Waveform 0**.

The pCLAMP software creates a plot that 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 to select **Append** in the Destination Option group.

## Preparing Data for Presentation

You are next going to create a Layout window that contains 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 display in the Analysis window, or using the Page Setup settings. You can specify three parameters to include in the comment line: the file name, the acquisition date and time and the comment.
3. Select **Page Setup** settings and clear the three parameters for the comment.
4. Select **Edit** > **Copy to Layout Window** to display a Layout window with 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 **Clipboard**. Select **View** > **Layout Window** and paste the waveform.
6. Return to the Graph window. The Trace vs. Trace graph can be customized in the Properties dialog that displays when you double-click, 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 are removed. When the graph suits your needs, copy it to the Layout window. You now have a figure that illustrates 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 the pCLAMP software functionality, this tutorial introduces artificial noise into a data file. Then you 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 you are going to make it a little more realistic. This tutorial works with a section of the file only:

1. Select **View** > **Data Display** > **Sweeps** and enter **20000** in the **Sweep Length** field.
2. Select the first sweep using the **View** > **Select Sweeps** command, or right-click in the data area.
3. 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 to select **Results Window**.
4. The maximum number of samples the pCLAMP software can transfer is 1,000,000. The sweeps are 20,000 samples long, so you can select **Full Trace** in the **Region to Transfer** field.
5. You want to transfer the first sweep only, so the **Trace Selection** group should report **All Visible Traces**.
6. Click **OK** and look at the Results window to find the first portion of the data on Sheet 14.

Now you can start creating the artificial noise. 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 the example. Keep this default here.
3. The Function options 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 the pCLAMP software fills the cells. Your selection is represented graphically in the dialog.
6. Since you are going to create a sine wave, you 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, use a variation of the function:

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

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

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

   The constant π is accessed by entering 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 you enter is:

   4 * sin(pi * x/10)

7. On the Results sheet, you find the time, in sampling interval steps, in the first column. Since this is the independent variable, you 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. You 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 click **Next** after you enter a name.
10. Now that you have the original signal in column B, and line frequency hum in columns C through F, you are going to add the components together. Select **Analyze > Column Arithmetic**.
11. In the Column Arithmetic dialog there are controls to enter an Expression. Click **Columns** to specify a list of columns using the “cA” notation explained in the dialog, or to select from the list. Click **Operator** to display a drop-down list of basic mathematical operators, while the **Function** button lists the functions supported by the pCLAMP software. 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. Click **Undo** to revert the most recent change 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, if the **Force Excluded Region To Zero** option is checked, else it remains unchanged.
12. To add all but the Time column and put the results out in an empty column, you should apply the following expression:
    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. You 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 you just saved into an Analysis window with **File** > **Open Data** and again set the file type to .atf.
16. The pCLAMP software 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 distance. 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-click menu.

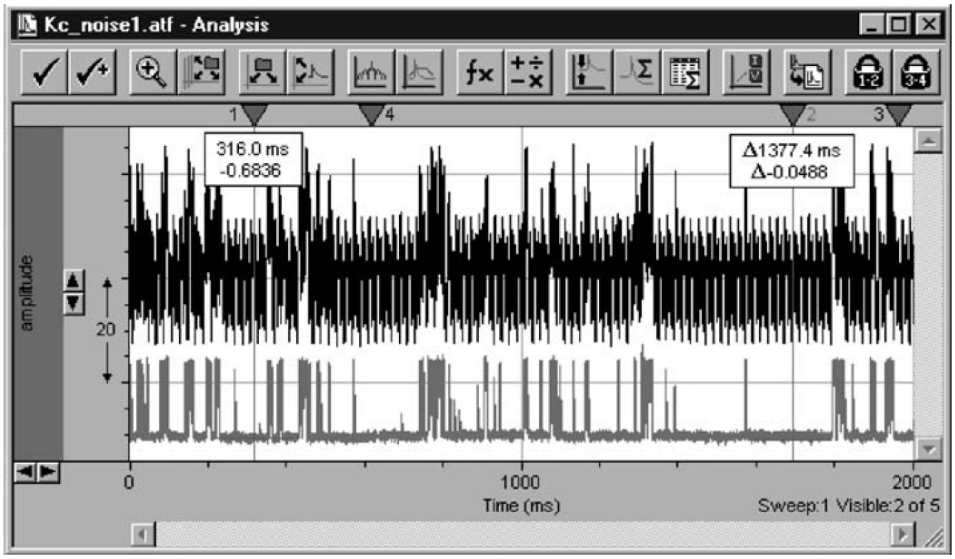



**Figure 7-2: Analysis window after Distribute Traces.**

19. The original recording 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, it is useful to perform a power spectrum before filtering:
    - Select **Sweep #6** only, **Autoscale** the **Y Axis** once more and go to **Analyze** > **Power Spectrum**.
    - Adjust the settings in the Power Spectrum dialog. See Clampfit Software Basic Analysis on page 135. For this tutorial, a **Window Type** > **Rectangular** together with the **Length** > **Maximum** will do. There is no need to Exclude spectral bins, and the analysis will apply to the full length of the Active trace. The pCLAMP software creates a Graph window with the power spectrum in a Log-Log plot.
    - 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.
    - 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 you to adjust the parameters. You know that there is no more than the fourth harmonic in the signal. The pCLAMP software determines which harmonics are present and only filters existing ones. So it is a matter of execution speed to restrict the search for harmonics to a realistic number.
    - 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.
    - The pCLAMP software can automatically determine the line frequency in the signal, but since it is known, set this parameter manually. After you click **OK**, the filter removes the hum, as you can convince yourself by performing another Power Spectrum and appending it to the existing plot.
    - The new power spectrum can be compared if you increase the line thickness of the plots, via the Graph Window Properties. 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. See Digital Filters on page 183.

If you have episodic data most of the steps you 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. You took a more complicated way to learn about the Results window, some of the features it offers, and the input and output of ASCII data. Furthermore, you learned how to use power spectra and digital filtering to improve the signal-to-noise ratio in the recordings.

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

In this tutorial you will learn how to use the pCLAMP software to separate the fast and slow components of a signal. This situation occurs 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, you 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 Digital Filters on page 183), which is not the case here with this older data file, as you can see if you select **File** > **Properties**. So you must convert the two sampling intervals of 200 and 600 µs to a single interval using either **Analyze** > **Data Reduction**, or **Analyze** > **Interpolation**. For comparison you 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 the pCLAMP software eliminates the excess samples.

- **Decimate** - Retains only every nth 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** - 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 you to select the Interpolation factor and one of two algorithms, Straight line or Cubic spline.

- **Straight Line** - Simply interconnects the existing samples, to create a step-like signal with newly introduced high-frequency components.
- **Cubic Spline** - Generates a smooth signal at the price of a slightly slower performance.

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

1. Open the Analyze > Interpolation dialog.

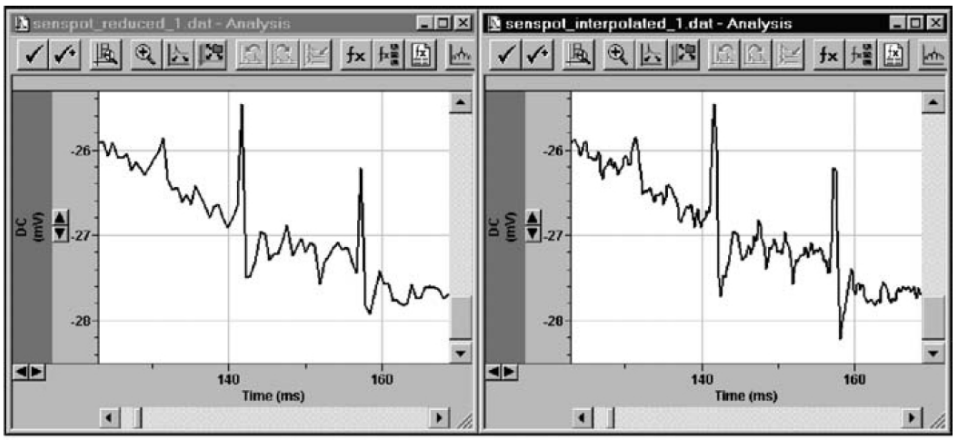



**Figure 7-3: Comparison of original (left) and interpolated (right) files.**

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 you intend to use software filters in the next steps, and the sampling rate determines the range of the allowable cutoff frequency, you 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 you want to isolate the slow component of the response, that is, the sensillar potential. Several types of lowpass filters are available. See Digital Filters on page 183.
4. Select 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 click **Select**.
8. In the Signals section of the Select Traces dialog, select **The Active Signal**. Since there is only one sweep in the file, skip selecting **Traces**.
9. Click **OK** and to report your selection in the Filter dialog.

10. Since you want to apply the filter to the entire length of the trace, set the **Region To Filter** to **Full Trace** and click **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. Type **Name** > **70 Hz Lo**.
3. Click **Save**.

How can you be sure now that 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

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

You want to compare the resulting trace to the existing signals in the file. Therefore, you 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**.

You are now ready to subtract the traces:

1. Open **Analyze** > **Arithmetic**.
2. Click **Traces** and select **Subtracted** from the **A** signal list.
3. Click **OK**.
4. Click **Traces** again and select **DC (mV)** from the **A** signal list.
5. Click **OK**.
6. From the **Operator** list select **- (subtract)**.
7. Click **Traces** again and select **70 Hz Lo (mV)** from the **A** signal list.
8. Click **OK**.

   The expression you then apply should resemble:

   T{VISIBLE} = DC:T{VISIBLE}-70 Hz Lo:T{VISIBLE}.
9. Click **OK**.

You can try other filter cutoff frequencies or filter types. The lower the cutoff frequency, 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. The more you increase the cutoff frequency, the larger are the residuals of the action potentials. The 70 Hz you originally chose was a reasonable compromise.

As you try other filter types, you will notice that the infinite impulse response filters (for example all types, but Gaussian and Boxcar) require a comparatively long time to phase in, and additionally start at 0 mV. In the following paragraphs of this tutorial you will learn two ways to deal with them. See Digital Filters on page 183.

After you isolate the sensillar potential, you will 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. You 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 **DC2 (mV)** signal to make it the active signal.
5. Select **Analyze** > **Filter**.
6. Select **Highpass**. Two types of Highpass filters are available: RC (single pole) and Bessel (8-pole).
7. Select **RC (single pole)** and specify **100 Hz** as the **–3 db cutoff** frequency.
8. Click **OK**.
9. 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.
10. To remove this transient:
    - Bring the cursors into the current region using **Tools** > **Cursors** > **Bring Cursors**.
    - Drag cursor 1 to the start of the file and cursor 2 to about 20 ms.
    - 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 allows you to specify the signals and sweeps to include.
    - Select **Fixed value** and enter 0.
    - In **Region to force**, select **Cursors 1..2**.
    - 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.

The other way to avoid the end effects is to remove 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 region
- **Slope** - Fits a regression line, then the result is subtracted from the entire trace, so that the region coincides with a straight baseline at y = 0.
- You can enter a fixed value, which is to be subtracted from the traces.
- You can correct an irregular baseline if you select **Adjust Manually** and click on the pink correction line to create break points that you can drag to follow an irregular curve.

Assuming the data values before the onset of the sensillar potential represent the baseline, the most suitable selection for your 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.

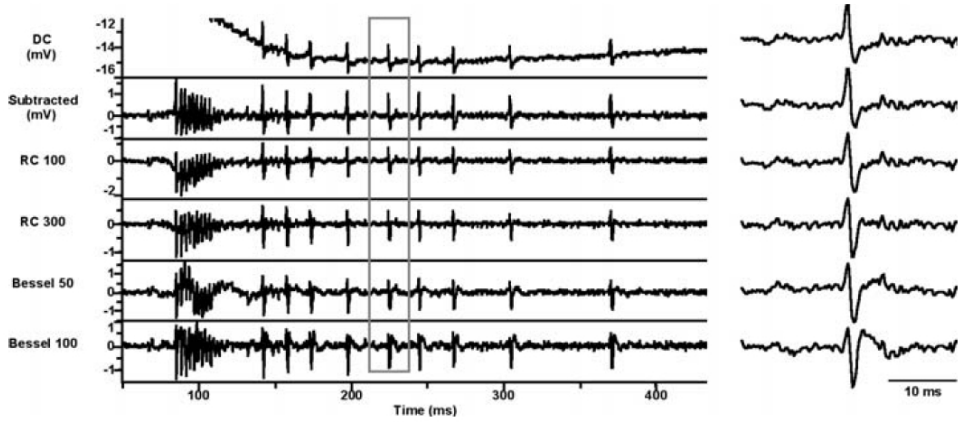



**Figure 7-4: Effects of different filter settings.**

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.

Now that you have separated the sensillar potential from the action potentials, you can evaluate 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**. You can use the cursor measurements to characterize the sensillar potential. Or you can 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**.
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**.

The pCLAMP software provides sophisticated fitting features, giving you the opportunity to select from a set of 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. See Curve Fitting on page 209.

A number of .atf files can be found in the ..\Program Files\Molecular Devices\pCLAMP11.x\Sample Data folder. Their file names indicate their functions and they are suited to help you become more familiar with the Fit feature.

You 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). You will test exponential functions of different orders and compare these models. For this purpose the pCLAMP software 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. Deselect **Select Fitting Methods Automatically** and select **Levenberg-Marquardt**, and **Weighting Method** > **None**.
3. On the Data/Options tab, enable **Compare Models**.
4. The **Starting Term** and **Ending Terms** are automatically set to the maximum possible; the 95% **Confidence Level** should be adequate.
5. On the Seed Values tab, review or alter the estimates the pCLAMP software has automatically generated. If you change the original values, they can be restored with the **Auto-Estimate** button.
6. Click **OK** to start the pCLAMP software fitting exponential functions of increasing order to the data. While the Automatic checkbox in the Compare Models group is selected, the comparison stops if a higher-order model does not improve the fit. If the Ending Term is set manually, the pCLAMP software continues up to the specified order.
7. Select **Analyze** > **Fitting Results**. The Parameters and the fit Statistics display on two tabs. The **Model Number** field at the bottom of the dialog allows you to review the values for the different models. In the Best Model group on the Statistics tab the pCLAMP software reports that, under the specified fitting conditions, a second order exponential function reasonably describes the fitted portion of the data.
8. Click **Copy to** > **Clipboard** or **Lab Book** to copy to the respective destinations the results for the model that displays 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 example of these signals are action potentials extracellularly recorded from insect sensilla (Frazier & Hanson 1986, Schnuch & Hansen 1990). This tutorial will demonstrate how the pCLAMP software can be used to characterize the waveforms in a series of similar signals. Then you will investigate whether the differences in the measured parameters permit us to assign them to different subpopulations, and finally you 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. The lower right corner of the Analysis window reports the number of the active sweep as **63**.
5. Open **View** > **Select Sweeps**.
6. Select **Sweep 63** and click **Invert**.
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. You 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, you can see that analog filters equally affect the waveform. Therefore, you remove the drift in the baseline using **Analyze** > **Adjust** > **Baseline**:

1. Use **Edit** > **Create Duplicate Signal** 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.
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-click menu.

11. Open the Analyze > Adjust > Baseline dialog and select **Adjust Manually**. Click **OK**.
12. A pink correction line displays which you can drag 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, click **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 you 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). The pCLAMP software 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. Use this option 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 you 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 you are going to determine waveform parameters that might be characteristic for the individual types of the action potentials. The measurements the pCLAMP software can perform on the data file are essentially the same that the Clampex Software offers during acquisition, so if the recordings are well-reproducible regarding time course and amplitude, it will save time if you set up the Clampex Software 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 your action potentials, not all pCLAMP software measurements are relevant. While the Peak amplitude is interesting, the Time of peak is identical for all sweeps, because you 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.
   After **OK** the pCLAMP software writes the measurements to the Statistics sheet in the Results window.

The tutorial assumes that the action potentials belong to two populations. First you will distribute them into two groups, evaluating their peak-to-peak amplitude. Then you will investigate whether the other parameters in these two groups are significantly different from each other.

In the Results window on the Statistics tab select the column named **R1S1 Time of Antipeak** by clicking on its header. Now select **Edit** > **Insert** > **Columns** to create a new column. Double-click on the new column 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, for example the time of the positive peaks after aligning, from the Antipeak column.

There are different ways to determine 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.

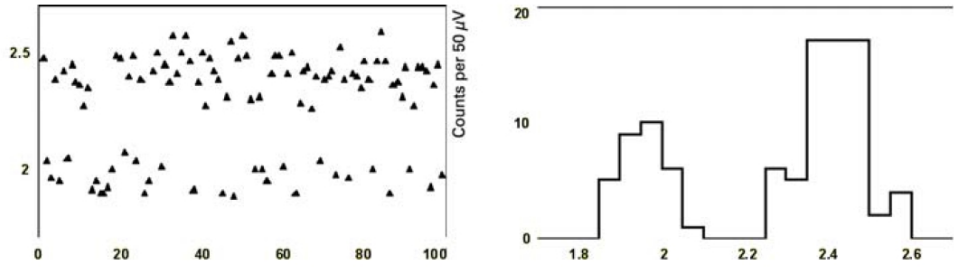



**Figure 7-5: Scatter plot and histogram showing groupings in peak-peak measurements.**

In this 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, only the **Number Per Category** is relevant for this project.

2. Select **Perform Breakdown Analysis**, select **Peak-Peak** in the **Category** column drop-down, check **Bin The Categories**, and click **Specify Bins**. A dialog displays where you specify bins of user-defined or fixed size. To cover the entire range of the occurring peak-to-peak amplitudes, specify **20 Fixed Bin Size** 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. The pCLAMP software writes the statistics on the Basic Stats sheet in the Results window.
4. Select the **Bin Center** column first and click the **X** toolbutton to make it the X column, which is reported in the column header. Then select the **#/Cat** column and click **Y+**.
5. Now go to **Analyze** > **Create Graph**, or click the toolbutton. Unless a different 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 you click **OK**, the plot should look similar to the histogram in Figure 7-5. Again there is no amplitude around 2.15 mV. So you 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, you will test this hypothesis.

The pCLAMP software features a number of statistical tests, which are accessible in the bottom section of the Analyze menu. For each of the parameters you determined, you 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 UTest, 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** 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 Columns 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 you use two user-defined bins, comprising the large and the small peak-to-peak amplitudes, respectively. Select **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 available when one of the bins in the Bin Number column is highlighted.

Upon OK, the Lab Book window displays to report a number of parameters for every column you 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 this 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 this 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 you 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 this 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 you have collected good evidence that the data file contains two types of action potentials with different amplitudes. For further evaluation you are going to save them in separate files now.

First you 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 highlight any other column in the drop-down.
2. The default **Row Range** depends on the selection, in this example it includes all non-empty rows. The **Columns To Sort** should 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:
   - Tile the Results and Analysis windows vertically, select the Analysis window and go to **View > Select Sweeps**.
   - In the **By Selection** field, highlight the sweeps whose numbers are in the **Trace** column now, all the time holding down the **<Ctrl>** key.
   - Before you click **OK**, select **By User-Entered** list and copy the contents of the **Range** field, which exactly reflects your selection, to the computer 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 be done if you click **<Ctrl>** before you start highlighting the remainder of the sweeps during the second turn. After you click **OK**, only the small action potentials display.
6. Save the file as *spikes_s.abf* 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 click **Invert**. 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.

# Chapter 8: Clampfit Software Basic Analysis


This chapter explains the basic analysis functionality in the Clampfit Software. For the optional advanced analysis information, see Clampfit Software Advanced Analysis on page 155.

## Fourier 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 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.

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. 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 pCLAMP Software on page 23.

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. 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 be a problem. Fortunately, there are procedures that can minimize such leakage. See Windowing on page 140.

## The Fourier Series

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 pCLAMP Software on page 23.

For a periodic digital signal x, the coefficients of the spectral distribution can be represented by the Fourier Series, which is defined by:

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

where, for samples x = $x_0$ to $x_{N-1}$, ai 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 The Fourier Transform on page 137.

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.

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 be a problem. Fortunately, there are procedures that can minimize such leakage. See Windowing on page 140.

## The Fourier Transform

Computing the Fourier Transform is a computationally intensive task that requires a great deal of redundant calculation. 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, this equation can be modified to apply to nonperiodic signals. Starting with a periodic signal of length N, you select an arbitrary number of points in the center of the signal. You then stretch the signal by adding zeros to either side of the selected points. Continue to add zeros until $N \rightarrow \infty$. At this point the neighboring repetitions have moved to $\pm\infty$ and you are left with a nonperiodic signal.

If the signal is stretched in this way then the coefficients, ak, 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. Think of this in terms of a continuous, rather than a discrete, spectral distribution.

As $N \rightarrow \infty$ the product $Na_k$ remains finite although each spectral coefficient, $a_k$, becomes vanishingly small. Write $Na_k$ as X. 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:

$$X = Na_k = \sum_{n=0}^{N-1} x[n]e^{-j\Omega n}$$

Since x[n] is now nonperiodic the limits of summation should be changed. Since x[n] exists for both positive and negative values of n you sum between $n = \pm\infty$. Write X as X(Ω) 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

The Fast Fourier Transform (FFT) is 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 Clampfit Software uses the Fast Fourier Transform (FFT) for spectral decomposition. 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 you have signal x[n] with N points where N is a power of 2 then you 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.

The pCLAMP software 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 (Pi) 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, fw, 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 units2, 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 is almost always expected. See The Fourier Series on page 136.

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.

The pCLAMP software 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 displays in the dialog 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 the pCLAMP software.

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 pCLAMP Software on page 23. The highest frequency component that can be resolved by Fourier analysis is only one-half the sampling frequency. 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 the pCLAMP software where the scale ranges from 0 to one-half the sampling frequency. A two-sided display would 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 given by:

$$W = f_s / L$$

## Membrane Test

The Membrane Test in the Clampex Software (**Tools** > **Membrane Test**) generates a definable voltage pulse and reads off a range of measurements from the current response:

- Membrane capacitance (Cm)
- Membrane resistance (Rm)
- Access resistance (Ra)
- 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 (for example 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 20% to 80% ordinates, or other ordinates, according to the settings in the Proportion of Peak to Fit section in the Configure > Membrane Test Setup dialog. The fit displays 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 ($C_m$).

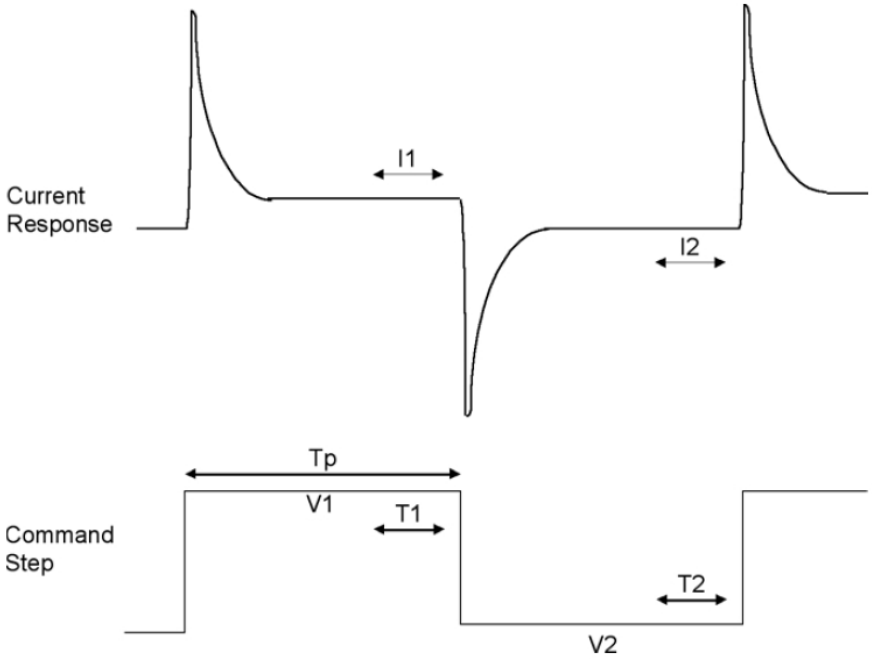



**Figure 8-1: Regions of current response used in Membrane Test calculations.**

The average current ($I_1$) is measured during period T1, 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 T2, 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 8-2).

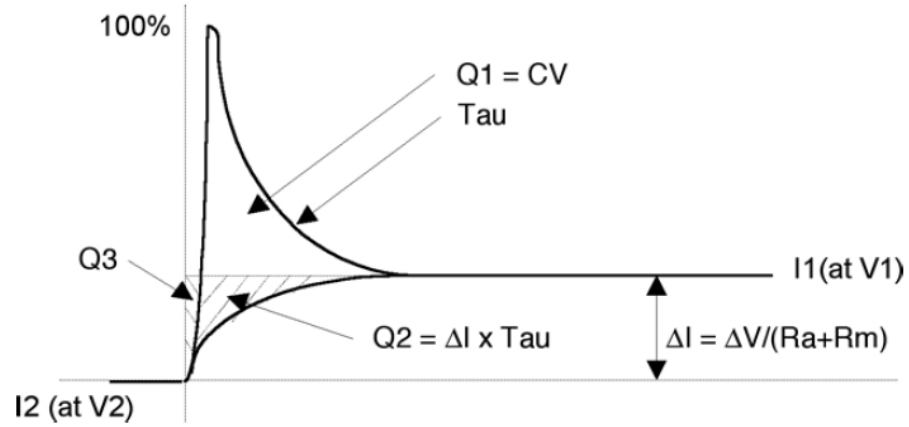



**Figure 8-2: Derivation of Membrane Test results.**

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 ($\tau$) 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 It before integrating.

$$Q_2 = \Delta I \times \tau$$

$$Q_t = Q_1 + Q_2$$

$C_m$ is derived from $Q_t = C_m \times \Delta V_m$

where $\Delta V_m$ is the amplitude of the voltage change across the capacitor, for example the change in the membrane potential. In the steady-state, the relation between $\Delta V_m$ and $\Delta V$ is:

$$\Delta V_m = \Delta V \times R_m / R_t = \Delta V \times R_m / (R_a + R_m)$$

where $R_t$, $R_m$ and $R_a$ are the total, membrane and access resistances, respectively. Substituting for $\Delta V_m$ from (2) into (1) derives:

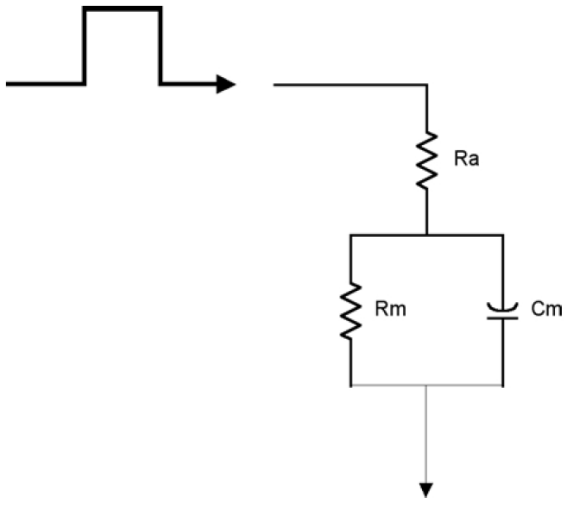



**Figure 8-3: Idealized cell circuit.**

The circuit elements $R_a$, $R_m$ and $C_m$ are readily derived as follows. Substituting (3) for Cm in the definition of the time constant = $R_a$ x $C_m$:

$$\tau = Q_t \times R_a / \Delta V$$

hence:

$$R_a = \tau \times \Delta V / Q_t$$

which provides access resistance directly as a function of measured variables. The total resistance is calculated from the steady-state response:

$$R_t = \Delta V / \Delta I = R_a + R_m$$

from which one obtains:

$$R_m = R_t - R_a$$

$C_m$ is then obtained from (3):

$$C_m = Q_t \times R_t / \Delta V \times R_m$$

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 the pCLAMP software, 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 can 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, 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.

See Clements and Bekkers, 1997.

## Single-Channel Event Amplitudes

Filtering data reduces the amplitude of rapidly occurring events. Short single-channel events have an attenuated amplitude. The points at the ends of long single-channel events have a smaller amplitude than in the absence of the filter.

When the software 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 considers the combination of digital filtering applied in the software prior to the single-channel search and the acquisition filters.

$$n_f = \frac{1}{2\pi\Delta t}\sqrt{\frac{1}{f_{Inst}^2} + \frac{1}{f_{CyberAmp}^2} + \frac{1}{f_{Postacq}^2}}$$

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 and CyberAmp filters used during acquisition, and $f_{Postacq}$ is the –3 dB cutoff frequency of the software 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 in the Results sheet State column.

**Note:** Failing to correctly specify the instrument filter you used when you acquire data invalidates the calculation of single-channel event amplitudes by the pCLAMP software.

### Level Updating in Single-Channel Searches

To accommodate baseline drift in single-channel detection, the search dialog allows you to specify automatic level updating, applied as a search proceeds. You can update the baseline without altering the relative amplitudes of the other levels, or update all levels independently.

When you select this you must set the amount that new events contribute to the level’s running amplitude. This weighting is entered as a percentage. With a level contribution of 10% 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:

$$(0.9 \times 10.0\ pA) + (0.1 \times 10.5\ pA) = 10.05\ pA$$

The algorithm includes a means to weight short events less, so that rapid events influence the level less than the level contribution you stipulate. This is because short events return amplitudes less than actually occurred (amplitudes of longer events are measured from the central portion of the event, to avoid the filter-affected transition periods at the start and end of the event). Because the reduced weighting of short events mitigates filtering effects, the amount of filtering a signal has undergone is used to determine the length of brief events, for example, 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, two time constants is the time equivalent of nf, for the number of samples, in the following equation:

$$n_f = \frac{1}{2\pi\Delta t}\sqrt{\frac{1}{f_{Inst}^2} + \frac{1}{f_{CyberAmp}^2} + \frac{1}{f_{Postacq}^2}}$$

Here, t is the sampling interval, and $f_{Inst}$, $f_{CyberAmp}$, and $f_{Postacq}$ the –3 dB cutoff frequencies of the instrument (for example amplifier), CyberAmp and pCLAMP software digital filters respectively.

For example, 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 these events make to the level update is directly proportional to the extent the event is shorter than this. For example, with a level contribution setting of 10%, an event lasting 8 ms (half a short event for these sampling and filtering rates) contributes 5% to the new level. All events over 16 ms affect the level update equally, at 10%.

## Kolmogorov-Smirnov Test

Use the two-sample Kolmogorov-Smirnov (K-S) test 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 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.

Use the K-S test 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). For comparing two different cumulative distribution functions, $S_{N1}(x)$ and $S_{N2}(x)$, the K-S statistic is:

$$D = \max_{(-\infty < x < +\infty)} \left| S_{N_1}(x) - S_{N_2}(x) \right|$$

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:

$$Q_{KS}(\lambda) = 2 \sum_{j=1}^{\infty} (-)^{j-1} e^{-2j^2\lambda^2}$$

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:

$$P(D > observed) = Q_{KS}\left(\left[\sqrt{N_e} + 0.12 + \frac{0.11}{\sqrt{N_e}}\right]D\right) \quad \text{(Stephens, 1970)}$$

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.

The pCLAMP software supports both one-dimensional and two-dimensional K-S tests. For 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 the following five locations in the pCLAMP software.

### Normalize Traces

Use Analyze > Normalize Traces 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 baseline region and $p_{|y|}$ is the largest absolute excursion between the mean baseline value and the peak region, where, for points n in the peak region:

$$p_{|y|} = max|y_n - b_y|$$

If the largest excursion from by 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 by 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. 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

Use Analyze > Normalize to normalize 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 can create 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 an 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:

$$y = (ix - x^2)/N$$

The free parameters *i* and *N* can be used to calculate the average synaptic parameters:

$$Q_w = i/(1 + CV_I^2)$$

and:

$$P_{rw} = x(i/N)(1 + CV_I^2)$$

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:

$$CV = SD/mean$$

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. The pCLAMP software 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:

$$y = ix$$

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 to choose 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, you are 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 interevent 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 intervals relative to a central event (for example,. 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. Tc 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 $T_{co}$ you have:

$$T_c = T_{co} - T_{co} \quad (= 0)$$

$$T_b = T_{bo} - T_{co}$$

$$T_a = T_{ao} - T_{co}$$

These data can display as a raster plot, where the data from each trace displays as individual points. A composite histogram having a bin width, using the times $T_b$ and $T_a$ from all of the data traces, can 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 the pCLAMP software. The central events must be tagged explicitly during event detection. This analysis cannot otherwise detect $T_c$. Data from other sources can 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 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:

$$T_o = \sum L t_o$$

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 be the case if the channel activity is low and/or the openings are brief. 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 $NP_o$, can be computed by:

$$NP_o = \frac{T_o}{T_o + T_c}$$

where $T_c$ is the total closed time.

# Chapter 9: Clampfit Software Advanced Analysis

Clampfit Software Advanced Analysis functionality requires an optional software license and a USB dongle.

## Batch Analysis

**Note:** You must install the MDC File Server to use Batch Analysis functionality. See Configuring MDC File Server on page 32.

Clampfit Batch Analysis

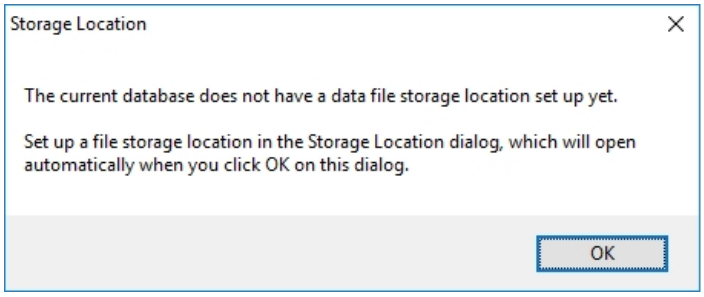
The following is how to use Batch Analysis and macros. There are several sample macros provided with the Clampfit Software Advance Analysis license. You can import corresponding sample data.

**Tip:** More usage information is available in the *Clampfit Batch Analysis Electrophysiology Data Management and Analysis Software User Guide*.

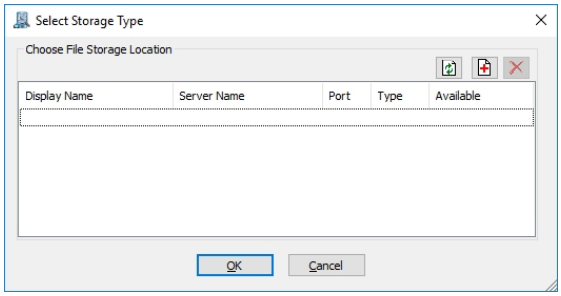
### First Time Use Database Configuration

If this is the first time you use the Clampfit Software Batch Analysis, do the following to configure the data storage folder:

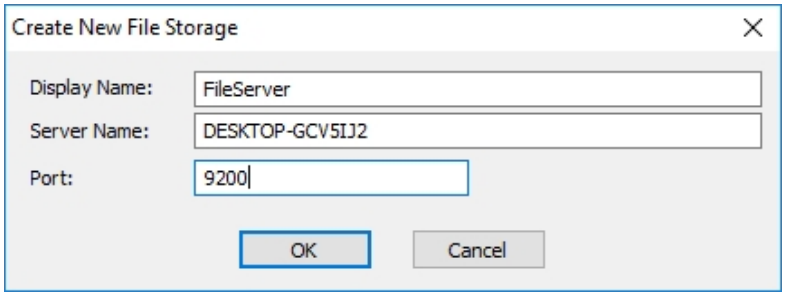
1. Ensure that the Clampfit Software license key dongle is installed on the computer that the software is installed.
2. Start the Clampfit Software.
3. Select **Batch Analysis** > **Run**.
4. When the Storage Location message displays, click **OK**.




5. When the Storage Location Type dialog displays, click  to display the Create New File Storage dialog.



6. To create a new file storage location:
   a. In the **Display Name** field, enter the name of the data file storage folder.
   b. In the **Server Name** field, verify that the local computer name is correct.
   c. In the **Port** field, leave the default setting as **9200**.
   d. Click **OK** to display the Storage Location Type dialog.




7. Confirm that **Yes** displays under the Available heading.
8. Click **Refresh** to update the list.
9. Click **OK**.

## Importing Data Files

The Clampfit Advanced Analysis software installer includes sample macros to familiarize yourself with the Batch Analysis functionality. To use these sample macros, you must import the sample data files. When you import your own data files they must be .abf files from the Clampex Software.

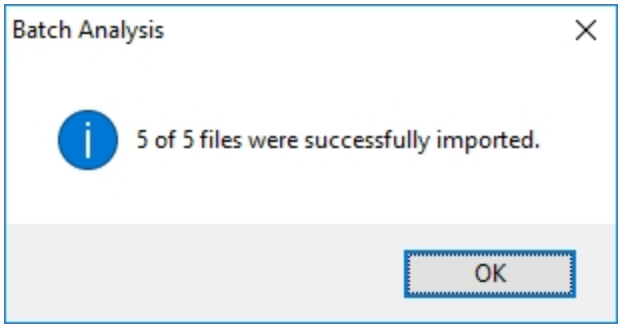
The following procedure uses the imported sample data Peak Data to use with the installed sample macro DemoPeak. These procedures apply to any imported sample data to use with any of the other corresponding sample macros.

To import data files:

1. In the Batch Analysis dialog, select **File** > **Import Data** > **Entire Folder**.
2. Navigate to **C:\Program Files\Molecular Devices\pCLAMP 11\Sample Macros\Peak Data**.
3. Click **Open**.

> **Tip:** Windows 10 users, if clicking Open fails, double-click on Peak Data.

4. On the confirmation message, click **OK**.

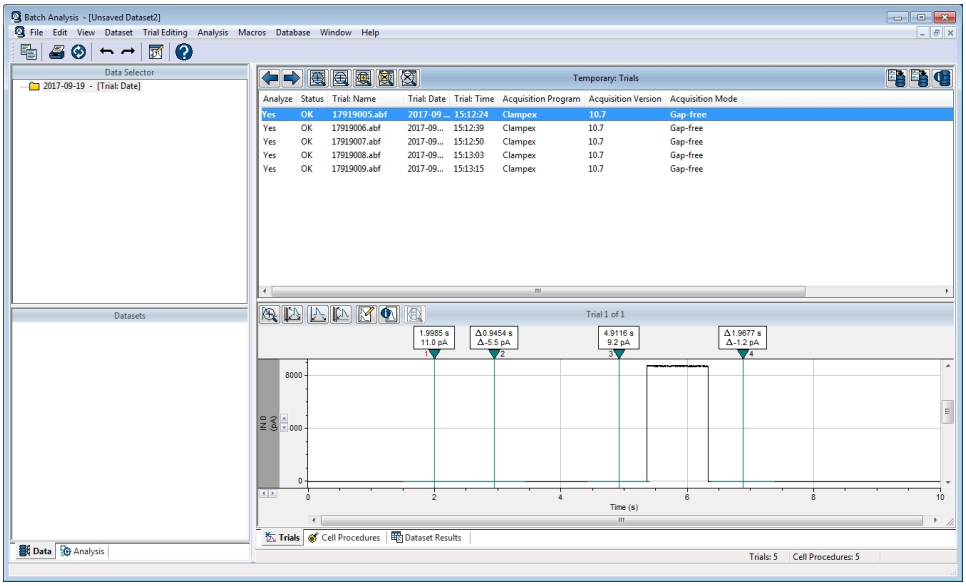


## Running Sample Macros

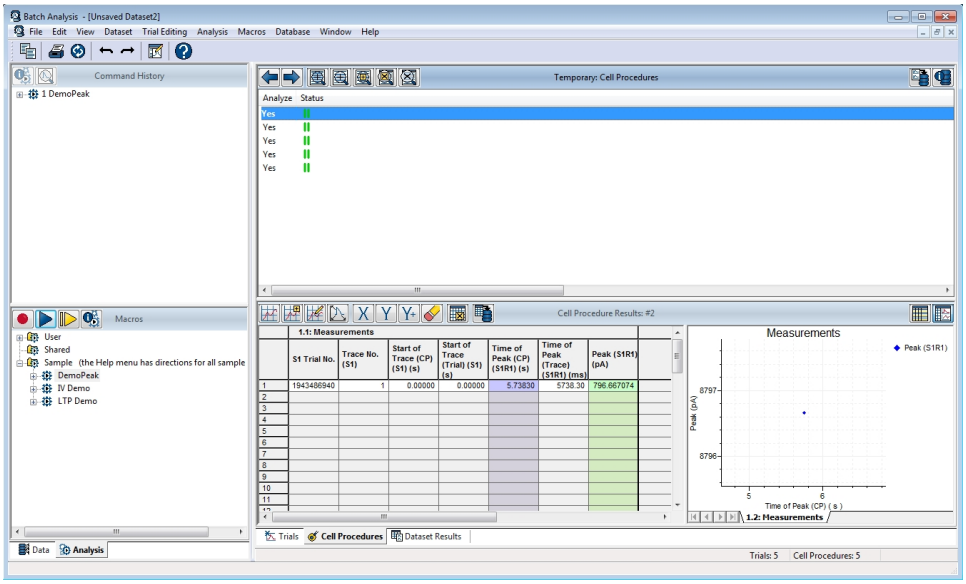
The following procedure uses the imported sample data Peak Data with the installed sample macro DemoPeak. See Importing Data Files on page 156.
These same procedures apply to any other imported sample data used with any of the other corresponding sample macros.

To run a sample macro:

1. In the top left quadrant of the Batch Analysis window under **Data Selector**, double-click the data folder you imported to load the Trials.




2. In the bottom left, select the **Analysis** tab.
3. In the **Macros** panel, in the **Sample** tree, right-click **DemoPeak** and select **Run Macro**.
When a macro runs successfully, the lower right panel displays the measurements table and associated graph.

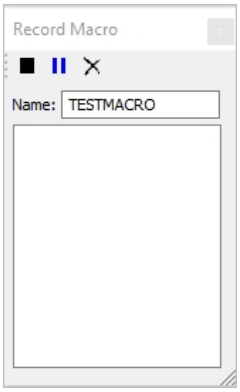


## Creating Macros

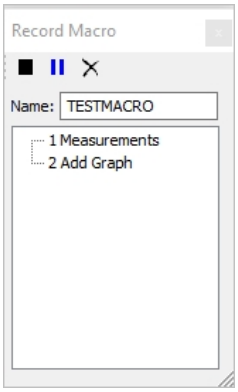
Any number of commands from the Analysis menu can be tied together as a single analysis sequence and stored in a macro. These new macros get stored in the Macros pane on the Analysis tab.

To create a macro:

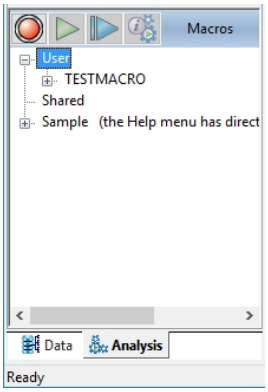
1. In the Batch Analysis dialog, select **Macros** > **Record** to display the Record Macro dialog.
2. In the **Name** field, enter the macro name. Do not press Enter.




3. Click the **Analysis** menu to select options and specify settings. The steps display in the Record Macro dialog as they are added.



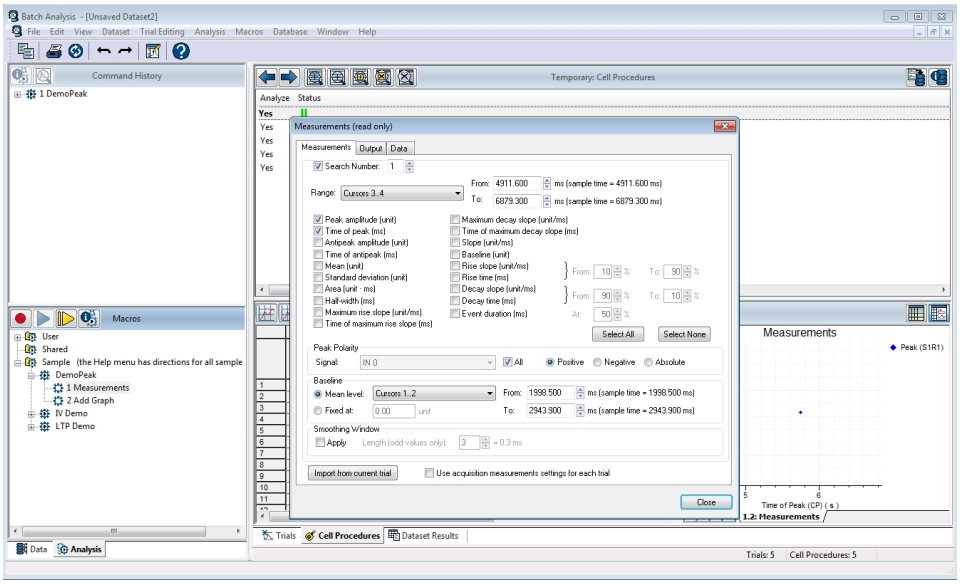

4. When you finish adding steps, in the Record Macro dialog, click **Stop** to save the macro and it displays in the Macros pane on the Analysis tab under User.



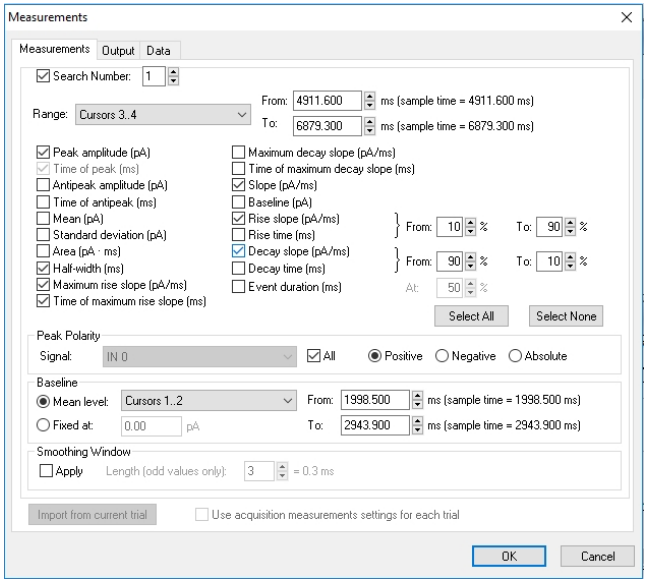

**Tip:** You can edit macros in the User tree.

## Modifying Sample Macros

Macros in the Sample tree are read-only. To view macro step settings, in the Macros pane on the Analysis tab, expand the macro name and double-click on a step.



To edit macros, right-click the sample macro name and select **Make a User Copy**. It is copied to the User macro tree, where you open the Analysis steps within the macro to reconfigure them. Click **OK** to save changes.




You can rename the macros and change the comments from the right-click menu.

You cannot add or remove steps to a macro when you edit it. To do this, record a new macro and use **Run Step** to copy the steps from the existing macro into the new one. Skip steps that are not wanted, or add new steps from the dialogs in the Analysis menu.

To delete a macro from the User tree, select the name and press **Delete**.

## Minis Search

The chemical version of communication between nerve cells relies on the release of a neurotransmitter from the presynaptic membrane. The chemical transmitter is encapsulated in synaptic vesicles (quanta) which, under specific conditions, bind to the presynaptic membrane and release their contents into the synaptic cleft. The transmitter molecules bind to receptors in the post-synaptic membrane, altering its permeability thereby promoting the movement of ions across the postsynaptic membrane. This postsynaptic ionic flow or current results in either membrane depolarization or hyperpolarization, depending on the type of chemical transmitter and the nature of the receptors which are permeable to specific ion types. During an action potential, one or two hundred quanta are released, generating a post-synaptic potential which spreads and either depolarizes or hyperpolarizes the neuronal cell membrane.

Miniature postsynaptic currents (mPSCs) or potentials (mPSPs) are called minis which are small spontaneous currents that occur when neurotransmitters are released from the presynaptic terminal without stimulation. The minis is thought to be a postsynaptic change in response to a spontaneous release of a single neurotransmitter vesicle. The minis could be excitatory and inhibitory depending on the type of neurotransmitter released from the vesicle and receptors on the postsynaptic site.

Use the Minis Search dockable panel (Search Definition dialog) to configure event searches based on template match criteria. The panel has fields to set signal noise, define event boundaries, curve fitting, and calculation of event statistics. A template file must be available for template matching. Minis Search is run on one data file at a time.

The amplitudes of minis are proportional to the size of the quanta. Measuring these amplitudes with microelectrodes thus provides a means of estimating the size of individual quanta. The analysis of such data provide insight into the function of neurons.

To use Minis Search you must have already created at least one template file suitable for the file you want to search, with the Event Detection > Create Template command.
The template defines a shape that is searched for in the file. Event amplitudes are not relevant to the finding of matches, but event durations are.

### Minis Search Data Recorded in Results Window

Each event that is accepted is reported on the Events sheet in the Results window, one line per event. Where a curve has been fitted, the measurements reported here are from the fitted curve and not the raw data.
The following data are reported in columns A–BG:

- **Trace** - The sweep number for the event.
- **Search** - The search number, depending on the number of restarts in the search session.
- **State** - Events found in a Minis search can have the following event states:
  - **A** - Accepted
  - **S** - Suppressed
  - **AT** - Accepted and Tag
  - **Added** - Added event
  - **Edited** - Edited event
- **Tag T** - Indicates the event was tagged when accepted.
- **Event Start Time** - Time after the start of the sweep that the event began.
- **Event End Time** - Time after the start of the sweep that the event finished.

- **Baseline** - Absolute baseline amplitude, as set. Other amplitude values reported are relative to this. If the baseline was set in the Analysis window all events in the sector will have the same baseline amplitude.
- **Peak Amp** - The amplitude of the point in the event that is furthest from the baseline, relative to the baseline amplitude.
- **Time to Peak** - Time from the start of the event to the peak.
- **Time of Peak** - Time of the peak point from the start of the sweep.
- **Antipeak Amp** - Amplitude of the point in the event furthest from the peak. Amplitude is relative to the baseline amplitude.
- **Time to Antipeak** - Time from the start of the event to the antipeak.
- **Time of Antipeak** - Time of the antipeak point from the start of the sweep.
- **Half-Width** - The period between the trace's crossing the half-amplitude point in its rising and decaying stages. The cross-over point nearest the peak is used in each case. Times at which the half-amplitude threshold is crossed may be calculated as falling between data points.
- **Half-Amplitude** - Half the peak amplitude, which is amplitude relative to the baseline.
- **Time To Rise Half-Amplitude** - Time from the start of the event to the half-amplitude crossing in the rising stage. The half-amplitude crossing nearest the peak is used.
- **Time To Decay Half-Amplitude** - Time from the start of the event to the half-amplitude crossing in the decaying stage. The half-amplitude crossing nearest the peak is used.
- **Rise Tau** - Time constant for the rising section of the event. There are three ways this value may be derived:
  - If no smoothing is selected, raw data points in the Calculate Taus From amplitude range are Chebyshev fitted with a single exponential equation and taus derived from this.
  - If Sum of exponentials smoothing is selected, the same procedure as for no smoothing is followed, except fitted data points are used instead of raw data points.
  - If Product of exponentials smoothing is selected, values are derived directly from the fit formulae.
  - If there are insufficient points in the nominated range for tau calculation, Not Found is reported.
- **Decay Tau** - Time constant for the decay section of the event. Derivation as for rise tau.
- **Max Rise Slope** -The maximum rate of amplitude increase, such as towards the peak, occurring between the event start and peak. If two sections of trace have the same increase rate the section nearest the peak is reported.
- **Time To Max Rise Slope** - Time from the start of the event to the midpoint between the two points during which the maximum rise occurred.
- **Max Decay Slope** - The maximum rate of amplitude decrease, such as away from the peak, occurring between the peak and event end. If two sections of trace have the same decay rate the section nearest the peak is reported.
- **Time To Max Decay Slope** - Time from the start of the event to the midpoint between the two points during which the maximum decay occurred.
- **Rise Slope** - Rate of increase in the rising stage of the event. The trace is searched from the peak to the event start. The first points falling below 80% and 20% of the baseline-to-peak amplitude range are found, and a linear regression applied to these and all intervening points.
- **Decay Slope** - Rate of decay in the decay stage of the event. The trace is searched from the peak to the event end. The first points falling below 80% and 20% of the baseline-to-peak amplitude range are found, and a linear regression applied to these and all intervening points.

- **Decay Time** - The time between the trace crossing 80% and 20% of the baseline-to-peak amplitude range in the decay stage of the event.
- **Area** - The area bound by the event data trace and the baseline. In the case of bipolar events the negative area is added to the positive area, so a total area of zero is possible. Area is calculated using simple summation of points within the event. For example, with an event from sample ten to sample sixteen, the sample-to-baseline amplitudes of each of the seven sample points in the range are summed, and multiplied by the sampling interval. Thus the area in the interval to the right of the final point is included.
- **Inst. Freq.** - Event frequency at the rate of the current and the previous event, such as the reciprocal of the Interval.
- **Interevent Interval** - The period between the peaks of the current and the previous event.
- **S.D. of Fit** - Where a curve has been fitted to the event, this column gives the standard deviation of the fit.
- **Template Match** - This value measures how well the event matches the template: the higher the value the better the match. This is the same parameter set in Template match threshold.

If graphs are configured for the search, the columns that follow column BG report data for these. The columns are in three groups:

- **G1–4**, **Midpoint & Bin Count**, **Cat 1–9** - These columns are used for histogram data. The graph number is given first (G1, etc.), then there are columns reporting the midpoint of each bin, and the number of points in each bin, for each level.
- **G1–4**, **ScPlot X & ScPlot Y**, **Cat 1–9** - These columns are used for scatter plot data. The graph number is given first, then columns for each of the selected X and Y parameters, for each level.
- **g1–4**, **ScPlot X & ScPlot Y**, **Cat 1–9 (Selected)** - These columns report the X and Y coordinates of points in a scatter plot that have been selected.

### Minis Search Dockable Panel

In Minis Search, please note that only one search category is required.

- **Restore Session** - Click to open the previously saved session.
- **Positive-Going / Negative-Going** - Select Positive-Going if searching for events with upward-pointing peaks, Negative-Going for events with downward-pointing peaks.

### Template Definition

Select and configure template settings in this group.

- **Template** - Select the search template for this category from the Load Template dialog box, opened with this button. The template displays in the template window below. Load Template is a standard file dialog.
- **Variable Amplitude Template** - Select this option in order that peak amplitude is included as a matching criterion in the detection of events.
- **Template Window** - A mini-Analysis window with many of the configuration options available for standard Analysis windows.
  The search template displays with two vertical analysis region cursors, and, depending on the **Set Baseline** selection below, one horizontal baseline marker.
  Drag the analysis region cursors to set event boundaries. The Clampfit Software searches for shapes similar to the whole of the template, but after these are found it takes all statistics, including start and end times, from the range you define with these cursors.
  If you select In Template in the Set Baseline group below the window, set the event baseline in this window by dragging the horizontal marker.
- **Set Baseline** - There are two options to set baselines.
  - **In Template** - The baseline is set in the template window by dragging the horizontal marker, or by entering a value in the field. This value is drawn from the scaling in the template window, which, by default, is not visible. Drag out the left-hand side of the window to view.
    This option links the baseline to the template. When events in the data file are matched to the template, the template is positioned with respect to amplitude as well as time, in order to get the closest match. The reported baseline then, for each event, can fall at a different absolute value.
  - **In Analysis Window** - The baseline is set in the Analysis window. Drag the horizontal marker in the Analysis window, or enter a value in the field.
    This option sets an absolute baseline that remains constant for all events that are found, for all search categories.
- **Bring Marker** - If the horizontal baseline marker is not visible in the template or Analysis window, depending on where you have elected to set the baseline. click this button to bring it into view.
- **Template Match Threshold** - The value entered here determines the degree of similarity an event must have to the template to be found in the search. The higher the value the more similar events need to be in order to be found, while a lower value increases the chance of false positives.
- **Min Half-Width** - Set a value to reject events with less than the value your enter.
- **Min Amplitude** - Set a value to reject events with less than the value you enter.
- **Rise Time** - The time between the trace crossing 20% and 80% of the baseline-to-peak amplitude range in the rising stage of the event.
- **Decay Time** - Set percentiles of peak
- **Rise Slope** - Set percentiles of peak.
- **Decay Slope** - Set percentiles of peak.
- **Show Slope Markers** - Select to show slope markers on events.

### Fitting

Enable Fitting to fit curves to found events. Event parameters reported in the Results window are then calculated from the curves, not the raw data. The standard deviation of the fit is also calculated. There are 2 fitting options:

- **Product Of Exponentials** - Select this option to apply exponential product fit curves to found events.
  The Calculate Taus control is not available when this option is selected, as the product of exponentials fit provides rise and decay tau values.
- **Sum Of Exponentials** - Sum of exponentials (two-term function only) between curves and are fitted to events.

#### Calculate taus from [ ] % of peak

Set the proportion of the rising and decaying phases of the trace, within an event, that is used to calculate rise and decay time-constants.

By considering only the midsections of rising and falling phases we can get a curve more amenable to calculation of tau values. The value entered with the spinner is a percentage of the amplitude range from the first value of the event to the peak value. This proportion of the trace is then removed from both the peak and baseline ends of the rise and decay phases, and the remaining section used for derivation of the time constants for each.

### Search Region

Select the signals and traces you want to search from the Select Traces dialog, opened with the Select button. The current selection is reported beside the button.

Define the time range you want to search in the list. You can search the Full trace, between cursors, or, for episodic stimulation files, a particular epoch.

## How to Configure Minis Searches

How to configure a Minis search:

1. Open the data file you want to search for events.
   It opens into an Analysis window.
   Data files are generally .abf files, but may also be .atf or .dat files.
2. Scale the window to maximize the signal along the Y axis.
3. Use the **Event Detection > Minis Search** command to open the Minis Search dockable panel.
   The **Event Detection** toolbar opens in the Analysis window.
4. Click **Restore Session** to open the previously saved session.
5. Select **Positive-Going** if searching for upward-pointing events, or **Negative-Going** for downward-pointing events.
6. Click **Template** to open a standard File dialog that should default to the Templates directory where your templates are saved. If not, navigate to the directory that contains them.
7. Select the template to use and click **Open** to display the template in the Template window with two vertical analysis region cursors and one horizontal baseline marker.
8. Drag the analysis region cursors to define event boundaries. The data trace searches for matches with the whole template, but event statistics are taken from the region within these boundaries.
9. In the Set Baseline group, choose to set the baseline relative to the template in the Template window or to have an absolute baseline set in the Analysis window.

10. Drag the baseline marker in either the template or Analysis window to the baseline level, or enter a numeric value in the field in the search panel.
11. This completes all the parameters you are required to set in the Template Definition group. Optional settings
    - Set a **Template Match Threshold** value. A higher value has the search find matches closer to the template. The default value, 4, generally provides reliable detection with few false positives.
    - Deselect **Variable Amplitude Template** near the top of the dialog to find events of similar amplitude to the template.
    - Set a **Min Half-Width** value to reject events with less than the value you enter.
    - Set a **Min Amplitude** value to reject events with less than the value you enter.
    - Set percentiles for **Rise Time**, **Decay Time**, **Rise Slope**, and **Decay Slope**.
    - Fit smoothing curves to the events. Event statistics are derived from these.

      **Note:** The Sum of exponentials is set to two-term function only.

    - Select the portion of rising and falling traces from which rise and fall time-constants are calculated.
12. In the **Search Region** group at the bottom of the panel, confirm that the signals and sweeps for the search are correct. If not, click **Select** and make a selection.
13. In the **Search Region** list, select the portion of the selected traces to search:
    - The whole trace
    - A cursor-defined region,
    - An epoch region
14. If you chose a cursor-defined region, drag the cursors into position in the file.
15. You are now ready to run the search.
    You can configure a second search category to run concurrently with the first. Return to the Category spinner at the top of the panel, select "2", and select the checkbox. Configure the search as before (from step 6).
16. Click **OK** to start the search

## Event Editor

Click an event to capture and display it in the Event Editor edit frame. It is best practice to click within the event rather than at the peak. Select a detected event or missed event either from within the data window or the Event Editor edit frame. You can edit the event to alter the baseline level, the peak level, the start and end times, and the peak time.

Position the horizontal baseline marker to specify the baseline level. If this marker is not visible, click **Bring Markers** to bring it into view.

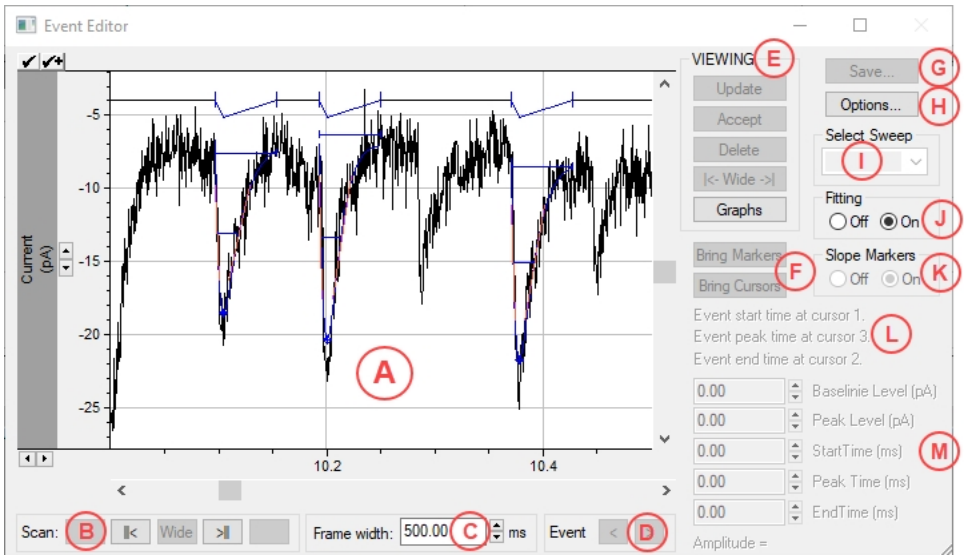
The peak level is initially positioned by adjusting the peak cursor. The horizontal peak marker locks to the peak cursor and is positioned at the data value you target by the cursor. After you position the peak cursor, the horizontal peak marker can be independently adjusted for fine tuning.

Adjust the corresponding cursors to specify the start and end times.

After an event is captured, click **Update** to execute the measurement of the event parameters. If the event was fitted, an outline of the fitted curve is superimposed over the event. This superimposition can then be visually assessed. If the results are satisfactory, click **Accept** to save the event. Otherwise, you can reset and update the cursors and markers as many times as necessary until you achieve a satisfactory measurement.

### Event Editor Dialog

The Event Editor dialog displays when you close the Minis Search dialog and the search completes.



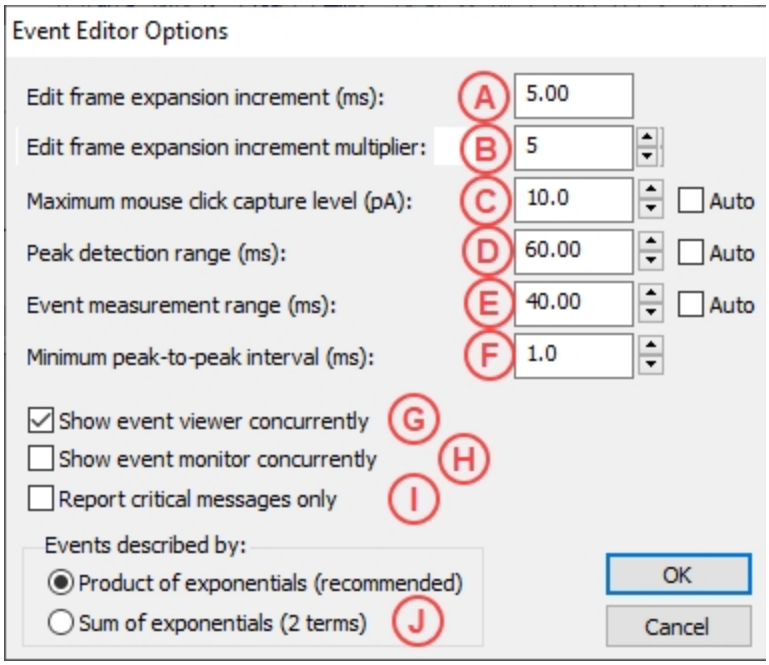

How to configure a minis analysis search:

1. **A Frame** - The Frame on the Event Editor displays a selected region of the data. The size of the Frame is determined by the Frame Width (C).
2. **B Scan** - The Scan group provides options that depend on the dialog state. In the Viewing state click **|I<** and **>I|** to move the edit frame (denoted by **|I|**) to the left or right by a distance equal to the edit frame width. This allows rapid scanning of the data trace.
3. **C Frame Width** - Enter a value or click the spinner to set the Frame Width. You can expand and contract this region by moving the edit region cursors in the main data window. The edit frame takes on two modes depending on the editing state. When editing an event, the width of the edit frame is narrowed to accommodate the width of the event being edited (narrow view). Otherwise, the editor frame takes on the width of the edit region in the data window (wide view).
4. **D Event** - Click the > to next sweep and the < to previous sweep in episodic file. Click the > to next event and < to previous event in gap-free file.
5. **E Viewing** - Viewing options:
   - **Update** - Click to update the measurements of a new or detected event as assigned in the edit window.
   - **Accept** - Click to save the edited event to the events list and displays the measurements in the Results window.
   - **Delete** - Click to remove an event from the event list and from the Results window.
   - **Wide** - Serves different functions depending on the editing state. In a narrow frame view (editing mode) this button cancels the event editing or resets the frame view to wide once you cancel the edit. The button text changes in accordance with the mode. When editing, this button reads **Cancel**. If you cancel the event edit, the button reads **|<- Wide ->|**. In Wide Edit Frame mode this button is not available.
   - **Graphs** - Calls the Graph setup dialog.
6. **F Bring MarkersBring Cursors** - Click **Bring Markers** to bring the horizontal baseline and peak markers into view.
   Click **Bring Cursors** to bring three event region cursors into view.

7. **G Save** - Click to save the data trace along with all edits to a file that is used to reconstruct the editing session at a later time.
8. **H Options** - Click to display the Event Editor Options dialog.
9. **I Select Sweep** - Click to select a sweep in an episodic data. This button is available when the data are from an episodic file.
10. **J Fitting** - When the events can be fitted, select **On** to compare the measurements obtained from the fitted curve or the raw data.
11. **K Slope Markers** - If the data are sufficient to accommodate the slope range as set in the definition dialog, rise and decay slopes are calculated. Select **On** to display the measured slopes in the edit frame.
12. **L Event Information** - This group displays the start, peak, and end times of the event when in narrow frame view.
13. **M Event Range** - Use these fields to set the new range for the event being edited. The five parameters are the Baseline Level, Peak Level, Start Time, End Time, and Peak Time. These values serve as the basis for the measurement of all parameters of the event. The event parameters also display in the Results Window.

## Event Editor Options

Use The Event Editor Options dialog to control event detection and views.




To define event editor options:

1. **A Edit Frame Expansion Increment** - Enter the number of milliseconds to have the Frame on the Event Editor dialog expand horizontally to encompass a wider view of the event being edited. This sets the amount by which the Frame is to expand. This is available only in the narrow edit frame view. The Scan buttons in the Event Editor dialog change their function and labels to reflect the edit frame expansion.
2. **B Edit Frame Expansion Increment Multiplier** - Use the spinner or enter the multiplication factor for the Frame expansion increment to increase the expansion amount.

3. **C Maximum Mouse Click Capture Level** - Use the spinner or enter the maximum mouse click capture level that sets the vertical range in the data window or edit frame within which to target a putative event. Select the **Auto** checkbox to have the software automatically set the maximum mouse click capture level.
4. **D Peak Detection Range** - Use the spinner or enter the peak detection range that sets the distance to either side of the mouse click position within which a search is conducted to find the peak of the data. If this region is too wide, then targeted lower amplitude peaks are replaced by any peak of greater amplitude within the peak detection range. Select the **Auto** checkbox to have the software automatically set the peak detection range.
5. **E Event Measurement Range** - Use the spinner or enter the event measurement range that sets the span within which the search for the event start and end times is conducted. This range should be wider than the anticipated width of the targeted event. After you estimate the event limits, the edit frame adjusts to confine the event. If this narrowing is too extensive, Click the **Span** buttons to expand the edit frame. Select the **Auto** checkbox to have the software automatically set the event measurement range.
6. **F Maximum Peak-To-Peak Interval** - Use the spinner or enter the maximum peak-to-peak interval.
7. **G Show Event Viewer Concurrently** - Select this checkbox to display both the Event Editor and an Event Viewer. The Event Viewer draws all selected events superimposed. When you click on an event in the Event Viewer the corresponding event is selected in the Event Editor, the Results window, and the Data window. You can save the events in the Event Viewer as an episodic .abf file if you require further specialized analysis.
8. **H Show Event Monitor Concurrently** - Select this checkbox to display both the Event Editor and an Event Monitor with some event parameters at the same time.
9. **I Report Critical Messages Only** - Select this checkbox to display only error messages. This suppresses warning messages.
10. **J Events Described By** - If an event is fitted, its parameters are derived from the fitted curve rather than the raw data, as is the case in the absence of fitting.
    - Select **Product Of Exponentials** to use the Product Of Exponentials function to fit events.
    - Select **Sum Of Exponentials** to use the Sum Of Exponentials two-term function to fit events.

## Population Spike Search

Population spikes are electrical signals that are generated by the synchronous firing of groups of closely spaced neurons that receive synaptic inputs simultaneously. The action potentials generated by this group of neurons produce a relatively large electrical potential (field potential) that can be easily detected by extracellular recording.

Distant to the focal site of the population spikes, a different signal is generated, which corresponds to a summation of the Excitatory Post-Synaptic Potentials (EPSP) resulting from the simultaneous release of neurotransmitter at the synaptic sites. This signal can also be detected by extracellular recording, and is characterized by a smaller initial potential difference preceding the population spike. This signal is referred to as a field excitatory post-synaptic potential (fepsp), and is mostly obscured by the population spike.

Also, a stimulus artifact is typically detected in the signal. This artifact is due to the propagation of an electrical stimulus focally applied to a region some distance from the recording electrode.

The components of the population spike signal are shown in Figure 9-1. In this recording, there are two occurrences of population spike firing spaced closely together.

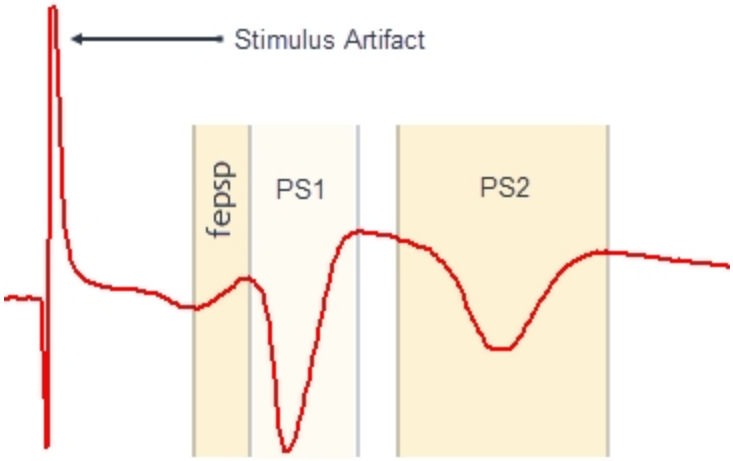



**Figure 9-1: The Population Spike and Field EPSP**

The event detection algorithm in Clampfit Software is optimized to detect and characterize population spike event with single or multiple peaks.

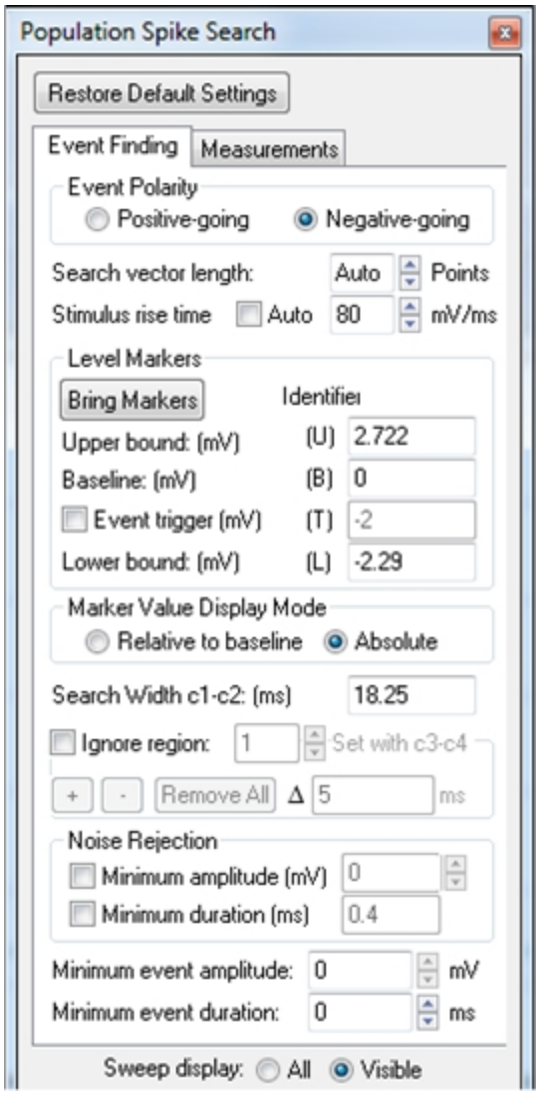
The Population Spike Search functionality enables you to set up specific search criteria and to select specific measurements.

The Population Spike Search feature is an option in the Event Detection menu. This option becomes available when a data file is imported.

The dockable tool panel contains the options for displaying the specified search region and a control for selecting one or more data traces for analysis. This tool panel can be docked at either side of the Analysis window by clicking on it and dragging it with your cursor. It can also be undocked and docked by double-clicking on it.

Tool panel has two tabs. The settings include:

### Event Finding Tab Settings




The Event Finding tab includes the options for configuring the detection parameters.

- **Event Polarity**:
  - **Positive-Going** - identifies a positive signal excursion with the “peak” located above the baseline (positive peak).
  - **Negative-Going** - identifies a negative signal excursion with the “peak” located below the baseline (negative peak). This is the typical choice for population spike analysis.
- **Search Vector Length** - sets the length in number of samples of positive-going and negative-going slope vectors that are used to analyze the contours and minima/maxima of the signal in conjunction with data point amplitudes and point-to-point slopes. This setting is at the center of the detection algorithm. Some experimentation with this setting might be necessary.
- **Stimulus Rise Time** - as needed, manually sets the stimulus rise time to a slower speed when the stimulus artifact is too slow for the default 80 mV/ms value.
  - **Auto** - sets the value to default 80 mV/ms.
- **Bring Markers** - displays the horizontal level markers.

- **Upper Bound** - sets horizontal marker **U**. Events over this level are ignored.
- **Baseline** - sets horizontal marker **B**. The level of no activity.
- **Event Trigger** - optional horizontal level marker. When selected for use, this level sets horizontal level marker **T**, which allows the segregation of events based on amplitude.
- **Lower Bound** - sets horizontal marker **L**. Events below this level are ignored.
- **Width c1-c2** - the temporal width of the search region. Display only.
- **Marker Value Display Mode**:
  - **Relative To Baseline** - has no effect on the detection routine. When selected, displays the absolute values as they appear on the vertical axis of the data window.
  - **Absolute** - has no effect on the detection routine. When selected, displays the values in the upper, lower and trigger level fields as the distance of excursion from the baseline.
- **Ignore Region** - sets cursors 3 and 4. These regions are identified as a dotted line, which can be seen when moving the cursors. The event search ignores points between cursors 3 and 4. You can specify any number of ignore regions. Use for fine-tuning the event detection.
- **Noise Rejection**:
  - **Minimum Amplitude** - events with amplitudes below this value are rejected as noise. This is different than the Minimum event amplitude option.
  - **Minimum Duration** - events with durations shorter than this value are rejected as noise. This is different than the Minimum event duration option.
- **Minimum Event Amplitude** - specifies the lower amplitude limit for the acceptance of an event.
- **Minimum Event Duration** - specifies the lower duration limit for the acceptance of an event.
- **Sweep Display**:
  - **All** - displays all of the visible and hidden traces in the Analysis window.
  - **Visible** - display only the visible traces in the Analysis window.
- **Search Region**:
  - **Select** - opens the Select Traces dialog. Specify which signal and traces should be applied to the analysis.

### Measurements Tab Settings

The Measurements tab includes those parameters that can be measured. These measurements are reported in the Results window. The data are arranged vertically by traces and horizontally by population spike events. The fEPSP rise times are available only in association with the first peak unless a multiple-pulse regime is analyzed.

**Note:** Selecting a measurement option does not guarantee that it will be measured. For example, if Latency is selected but the stimulus artifact is not included in the search region then latency cannot be measured and the Results table cell will be empty.

At least one of the following option must be selected:

- **Population Spike**:
  - **Rise Time** - time to rising phase over the specified range.
  - **Decay Time** - time to decay over the specified range.
  - **Rise Slope** - rate of rise over the specified range.
  - **Decay Slope** - rate of decay over the specified range.
- **Field EPSP**:
  - **Rise Time** - time to rising phase over the specified range.
  - **Rise Slope** - rate of rise over the specified range.
- **Start Time** - time of the start of the population spike event relative to the start of the recording.
- **End Time** - time of the end of the population spike event relative to the start of the recording. Includes all detected events within the search region.
- **Latency** - time from the largest stimulus artifact peak (usually positive) to the pop spike peak (Z-amplitude level).
- **Z-Amplitude** - amplitude of the population spike event, defined as the distance from a point on the tangent line between the ridge values to the peak amplitude.
- **Peak To Peak Time** - time between successive peaks in a burst of population spikes.
- **Area** - area, in mV-ms, of the event calculated between the ridge tangent line and the event profile.
- **Half Width** - duration of the event at the 50% amplitude level.
- **Coastline** - summation of point-to-point amplitudes over the event excursion, not including the ridge tangent line.
- **Sweep Display**:
  - **All** - displays all of the visible and hidden traces in the Analysis window.
  - **Visible** - display only the visible traces in the Analysis window.
- **Search Region**:
  - **Select** - opens the Select Traces dialog. Specify which signal and traces should be applied to the analysis.

## How To Configure Population Spike Searches

Data files can be searched for population spike events. The following procedure gives instructions to configure a search.

How to configure a population spike search:

1. Open the data file to analyze. It opens in an Analysis window.
   Data files are generally .abf files, but might also be .atf or .dat files.
2. Use the **Event Detection > Action Potential Search** command to open the Population Spike Search dockable panel.

   **Note:** The Event Detection toolbar opens in the Analysis window.
3. Select the **Event Finding** tab.
4. In the **Event Polarity** section, specify the polarity of the population spike. The correct setting of the polarity is critical to the search. Typically, population spike analysis uses Negative-Going polarity.
5. Set the **Search Vector Length** in number of samples of positive-going and negative-going slope vectors. In conjunction with data point amplitudes and point-to-point slopes, these vectors are used to examine the contours and minima/maxima of the signal. If these vectors are too long, they inaccurately reflect the local curvature. If they are too short, then the influence of noise becomes significant.
6. In the Level Markers section, click **Bring Markers** to display the horizontal level markers if they are not in view.
7. To set the search region, you must set the **Upper Bound (U)**, **Baseline (B)**, and **Lower Bound (L)** horizontal level markers. If needed select and set the **Event Trigger (T)** level. Only those events whose positive or negative peak crosses the trigger level are accepted.

   **Note:** The baseline is not used for event measurements but its position is used when estimating a baseline noise level, so it should be accurately set.
8. In the Marker Value Display Mode section, set values to display either as **Absolute Values** or **Values Relative To Baseline**. This setting does not affect the event detection.
   - **Relative To Baseline** - Has no effect on the detection routine. When selected, displays the absolute values as they appear on the vertical axis of the Data window.
   - **Absolute** - Has no effect on the detection routine. When selected, displays the values in the upper, lower and trigger level fields as the distance of excursion from the baseline.
9. Set the **Search Width c1-c2**.
10. Select **Ignore Regions**, set the region using cursors 3 and 4, and then to assign the **Ignore Region**, click **+**.

    **Tip:** In Population Spike Analysis, there is often an initial short, low amplitude event that represents the activation of presynaptic fibers. This event is not of interest, so enclose it in a region that is ignored.

    - To add an Ignore Region, increment the region numbers and set cursors 3 and 4 positions before clicking **+**.

      **Tip:** There is no limit to the number of Ignore Regions you can assign.
    - To remove an Ignore Region, select the region number, and then click **-**.
    - To clear all ignored regions, click **Remove All**.

11. Set **Noise Rejection** to specify the minimum amplitude and duration to accept an event metric. For example, an excursion of less than the noise rejection settings is ignored during the search for a maximum value (peak) of an event.
12. Set the **Minimum Event Amplitude**. Events with amplitudes below this value are rejected.
13. Set the **Minimum Event Duration**. Events with durations below this value are rejected.
14. For **Sweep Display**, select either **All** or **Visible**.
15. Select the **Measurements** tab.
16. Select the measurements and set the values to display at the end of the search in the Results window.
17. To start the search:
    a. Click **Select**.
    b. In the Select Traces dialog, make the selections.
    c. Click **OK**.
    d. When the search finishes, click either **Accept**, **Reject**, or **Enable Editing** for each of the detected events.

    **Tip:** You cannot make edits after you finish the search. See Editing Population Spike Search on page 175.

    The Event Viewer displays the entire set of accepted traces, which can be saved to a separate file.
    When the analysis completes and the events are accepted, the results display in the Results window.

### Editing Population Spike Search

If your detected population spike events need editing, it must be done during the active search process.

To adjust a detected event:

1. When you click **OK** to run the search and the search finishes, you must click either **Accept**, **Reject**, or **Enable Editing** for each of the detected events.
2. To begin editing, click **Enable Editing**.
3. Click on the trace close to the region you want to edit and the editing cursor appears.
4. To move the editing cursor, you can do one of the following:

   **Note:** The ridge values of editing are limited. You cannot push farther than the peak value nor the left ridge value of the next event.

   - On the screen, click and drag the editing cursor line to a new position.
   - Using your keyboard, press the right or left arrow keys to move the editing cursor line to a new position one position at a time.
   - Using your keyboard, press <shift> +right arrow or<shift> + left arrow keys to move the editing cursor line to a new position by several positions at a time.

     **Tip:** To undo editing, in the toolbar, click **Reset**.

5. When you are finished moving the cursor line, in the toolbar, click **Accept Edit and Resume**.

   **Tip:** Alternatively, you can left-click anywhere outside of the search region to accept the edited event.

6. Repeat steps 3 through 5 as needed.
7. To finish your editing, in the toolbar, click **Accept** and the next detected event appears.

   **Tip:** To go scroll through the detected event, on your keyboard press the **<** and **>** keys as needed.

To exclude a region of an event:

1. If there is a region in the detected event that should have been ignored, on the graph, click on the region and the editing cursor appears.
2. Right-click on the editing cursor line and select **Exclude Event**.

   **Tip:** To undo an Exclude Event, from the menu, select Event Detection > Restore Excluded Event.

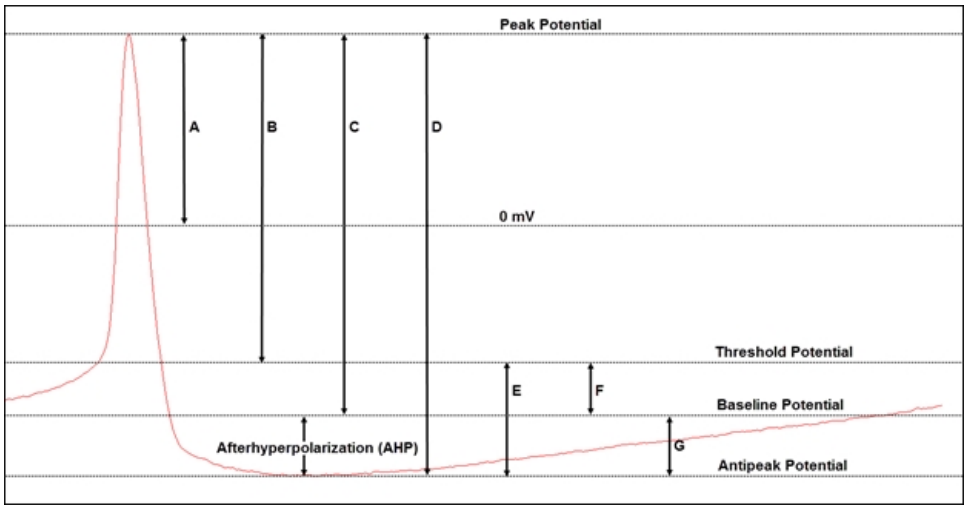
To quit editing:

In the toolbar, click **Quick Event Editing**.

## Action Potential Search

Action potentials are electrochemical manifestations of excitable cells. Important information can be gleaned from the characteristics of action potentials. To this end, Clampfit Software provides the tools for measuring several important properties of cardiac, sinoatrial and fast neuronal action potentials.

The Action Potential Search functionality enables you to select measurements such as peak amplitude, action potential duration, peak to peak frequency and time, and threshold potential. This feature is an option in the Event Detection menu. This option becomes available when a data file is imported.




A. Overshoot amplitude

B. Amplitude relative to threshold potential

C. Amplitude relative to baseline potential

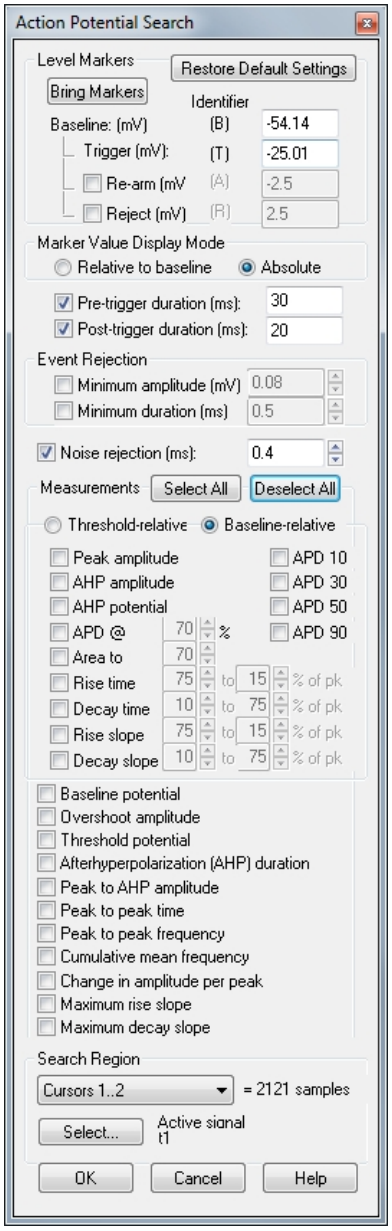
D. Amplitude relative to antipeak potential (minimum AHP potential)

E. Threshold potential relative to antipeak potential

F. Threshold potential relative to baseline potential

G. Baseline potential relative to antipeak potential

The dockable tool panel contains the options for detection and measurement of the action potential waveforms. This tool panel can be docked at either side of the Analysis window by clicking on it and dragging it with your cursor. It can also be undocked and docked by double-clicking on it.



Tool panel settings include:

- **Level Markers**:
  - **Bring Markers** - brings the horizontal level markers into view.
  - **Baseline** - required horizontal level marker. The baseline serves as a reference point for several measurements, therefore its position is critical.
    - **Trigger** - required horizontal level marker. When this level is crossed in both directions, the event is accepted for processing.
    - **Re-Arm** - optional horizontal level marker. When selected for use, this level signals the end of one event and a restarting of the search for the next event from that point. The information between the time of the re-arm level and the time of the next detected event is ignored. For example, this marker is useful if only the peak amplitude or frequency is of interest.
    - **Reject** - optional horizontal level marker. When selected for use, this level sets the upper limit for event acceptance. Events that cross this level are rejected. For example, this marker can be used to reject stimulus artifacts that contaminate the data.
- **Marker Value Display Mode**:
  - **Relative To Baseline** - has no effect on the detection routine. When selected, displays the absolute values as they appear on the vertical axis of the data window.
  - **Absolute** - has no effect on the detection routine. When selected, displays the values in the upper, lower and trigger level fields as the distance of excursion from the baseline.
- **Pre-Trigger Duration** - when selected for use, specifies the extent of the search region before the point at which the event was triggered. Affects the measurement of action potential properties that occur before the action potential peak. When not selected for use, the search routine uses the baseline crossing and/or detection of a preceding peak to indicate the beginning of the search region.
- **Post-Trigger Duration** - when selected for use, specifies the extent of the search region after the point at which the event was triggered. Affects the measurement of action potential properties that occur after the action potential peak. When not selected for use, the search routine uses the baseline crossing and/or detection of a subsequent peak to indicate the end of the search region.

  **Note:** When this option is off, the search continues until either the baseline is crossed or the next peak is detected.

- **Event Rejection**:
  - **Minimum Amplitude** - specifies the lower amplitude limit for the acceptance of an event. Peak amplitudes measurements at the trigger level that are below the specified value are rejected.
  - **Minimum Duration** - specifies the lower duration limit for the acceptance of an event. Evaluated at the APD50 level because this is the parameter that is most likely to be detected and accurately measured. Since Action Potential Duration (APD) values are measured relative to the baseline, if the baseline is not crossed, as might be the case during trains, the values below APD50 might not be detected.
- **Noise Rejection** - sets the minimum duration for acceptance of an event, as measured between the baseline crossings in milliseconds. Events shorter than the noise rejection level are excluded.

- **Measurements** - a group of measurements that are reported in the **Results** window. At least one measurement option must be selected.
  Choose one of the following relative display options:
  - **Threshold-Relative** - relative measurement selections display relative to their position between the peak and the threshold value.
    For example, the peak amplitude value displays between the peak and the action potential threshold on the graph.
  - **Baseline-Relative** - relative measurement selections display relative to their position between the peak and the baseline value.
    For example, the peak amplitude value displays between the peak and baseline level on the graph.

  The following relative measurement options are available:
  - **Peak Amplitude** - amplitude between the peak and the baseline level and the amplitude between the peak and the threshold potential.
  - **AHP Amplitude** - amplitude between the AHP potential and the baseline level and the amplitude between the AHP and the threshold potential.
  - **AHP Potential** - minimum value of the after hyperpolarization.
  - **APD @** - Action Potential Duration at percentage of your custom specification.
  - **Area To** - computed relative to the specified from peak value.
  - **Rise Time** - time to rising phase of the action potential over the specified range.
  - **Decay Time** - time to decay of the action potential over the specified range.
  - **Rise Slope** - rate of rise of the action potential over the specified range.
  - **Decay Slope** - rate of decay of the action potential over the specified range.
  - **APD 10** - Action Potential Duration at 10%
  - **APD 30** - Action Potential Duration at 30%
  - **APD 50** - Action Potential Duration at 50%
  - **APD 90** - Action Potential Duration at 90%

  The following absolute measurement options are available:
  - **Baseline Potential** - potential as set in Baseline.
  - **Overshoot Amplitude** - the amplitude between zero membrane potential difference and the peak potential of the action potential.
  - **Threshold Potential** - amplitude at which the action potential is initiated. An approximation of this position is found by finding the intersection of the rising phase of the action potential and the slope leading up to its initiation. Subsequently, a three-point tangent slope vector is used to find the position in the initial region where the slope is at or above 10 mV/ms.
  - **Afterhyperpolarization (AHP) Duration** - duration of the after hyperpolarization potential as measured between the downward and upward baseline potential crossings.
  - **Peak To AHP Amplitude** - amplitude as measured from the peak amplitude to the most negative value of the afterhyperpolarization.
  - **Peak To Peak Time** - time between successive peaks in a burst of action potentials.
  - **Peak To Peak Frequency** - frequency of successive peaks in a burst of action potentials.

- **Cumulative Mean Frequency** - frequency between the first event and each successive event.
  For example, if there are five events, the mean frequency of events 1-2, 1-2-3, 1-2-3-4, and 1-2-3-4-5 display. The last value, yields the mean frequency of events in the entire record.
- **Change In Amplitude Per Peak** - change in amplitude, relative to the baseline, of successive peaks in a burst.
- **Maximum Rise Slope** - largest positive slope of the event between the baseline and the peak as measured between every two successive data points in the rising phase of the action potential.
- **Maximum Decay Slope** - largest negative slope of the event between the peak and the baseline as measured between every two successive data points in the decay phase of the action potential.

- **Search Region** - used to specify the range over which the detection should take place. Events outside the specified search region are ignored.
  - **Select** - opens the Select Traces dialog. Specify which signal and traces should be applied to the analysis.

## How To Configure Action Potential Searches

Data files can be searched for action potential events. The following procedure gives instructions to configure a search.

How to configure an action potential search:

1. Open the data file to analyze. It opens into an Analysis window.
   Data files are generally .abf files, but might also be .dat files.
2. Open the Action Potential Search panel.

   **Note:** The Event Detection toolbar opens in the Analysis window as well.

3. In the **Level Markers** section, click **Bring Markers** to show the horizontal level markers.
   Two horizontal markers display in the Analysis window. The Baseline marker is labeled **B** and the Trigger marker is labeled **T**.
4. To set the Baseline, either:

   **Note:** All measurements are made relative to this setting.

   - Drag the Baseline **Identifier (B)** to the baseline level.
   - Set a numerical baseline value in the **Baseline** field in the search panel.
5. To set the Trigger, either:
   - Drag the **Identifier (T)** to a suitable level.
   - Set a numerical trigger value, relative to the Baseline, in the **To Trigger** field.
6. Set the trigger level. This is used to detect an event. The signal must cross the trigger potential in both directions in order to be accepted as an event.
7. Set the **Re-Arm** level (**Identifier (A)**), which signals the end of one event and a resumption of the search for the next event. The information between time of the re-arm level and the time of the next detected event will be ignored.

8. Set the **Reject** level (**Identifier (R)**), which specifies the maximum level for an event to be accepted. If the signal crosses the reject level, the event search begins again. Any events that have an amplitude above the rejection level are excluded.

   **Tip:** The Re-Arm and Reject markers appear only when the corresponding checkbox is selected.

9. In the **Marker Value Display Mode** section, set values to display either as **Absolute Values** or **Values Relative To Baseline**. This setting does not affect the event detection.
   - **Relative To Baseline** - Has no effect on the detection routine. When selected, displays the absolute values as they appear on the vertical axis of the data window.
   - **Absolute** - Has no effect on the detection routine. When selected, displays the values in the upper, lower and trigger level fields as the distance of excursion from the baseline.
10. Set **Pre-Trigger Duration** and **Post-Trigger Duration** to determine the range of data to analyze before and after an event is detected by trigger level crossing. Only data within these time spans and within the threshold limits is used.

    **Tip:** Pre-Trigger Duration and Post-Trigger Duration settings are used when the corresponding checkbox is selected.

    When these options are not used, the event analysis range is determined by baseline crossings, either in one or both directions, and/or subsequent trigger level crossings, such that an event end is signaled by the detection of a subsequent event.

11. Set **Noise Rejection** to specify the Minimum Amplitude and Minimum Duration to accept an event measurement. An excursion of less than the noise rejection settings is ignored during the search for a maximum value (peak) of an event.

    **Tip:** The Minimum Amplitude and Minimum Duration settings are used when the corresponding checkbox is selected.

12. In the **Measurements** section, select which measurements of the action potential to appear in the Results window.

    **Tip:** An approximation of the position of the Threshold Potential is found by finding the intersection of the rising phase of the action potential and the slope leading up to its initiation. A three-point tangent slope vector is used to find the position in the initial region where the slope is at or above 10 mV/ms.

13. In the **Search Region** at the bottom of the panel, in the drop-down list, select how much of the traces to search:
    - The full trace
    - A cursor-defined region,
    - An epoch region (if the file was recorded under a stimulus waveform)

    **Tip:** Events outside the specified search region are ignored.

14. Confirm that the signals and sweeps selected for the search are correct. If not, click **Select** and make a selection.

15. Click **OK** to save the search settings, start the search, close the configuration panel, and open the Event Viewer window.
    The peak position and duration marker is positioned above the detected event, and the slope markers are positioned. These event markers display as dotted lines.
16. Review the dotted line event markers and click **Accept** or **Reject** in the toolbar.
    Accepted event markers change to solid blue lines, the accepted event displays in the Event Viewer, and the next event is detected and marked. The slope markers display as thicker darker blue lines.

> **Tip:** Click **Nonstop** in the toolbar to detect and accept all subsequent events until the end of the search region is reached.

The Measurements you selected in the control panel display in the Results window.

# Chapter 10: 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 the pCLAMP software, 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 are instances when it is necessary to filter experimental data after they have been digitized. For example, to remove random noise or line frequency interference from the signal of interest. The pCLAMP software offers several types of digital filters to do this.

Lowpass filters include:

- Bessel Lowpass Filter (8 Pole) Specifications on page 186
- Boxcar Smoothing Filter Specifications on page 188
- Butterworth Lowpass Filter (8 Pole) Specifications on page 189
- Chebyshev Lowpass Filter (8 Pole) Specifications on page 191
- Gaussian Lowpass Filter Specifications on page 193
- RC Lowpass Filter (Single Pole) Specifications on page 195
- RC Lowpass Filter (8 Pole) Specifications on page 196

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 Finite vs. Infinite Impulse Response Filters on page 185.

Highpass filters include:

- Bessel Highpass Filter (8-Pole Analog) Specifications on page 199
- RC Highpass Filter (Single Pole) Specifications on page 198

The notch filter is available to reject a narrow band of frequencies. See Notch Filter (2 Pole) Specifications on page 200.

The electrical interference filter is provided to reject 50 or 60 Hz line frequencies and their harmonics. See The Electrical Interference Filter on page 201.

## Digital Filter Characteristics

An ideal filter has a rectangular magnitude response with no attenuation in the passband and full attenuation in the stopband. Ideal filters are *noncausal* in that the present output depends on future values of the input. They are not realizable. Realizable digital filters can approximate ideal filters in that the output can be delayed for a finite interval until all of the inputs have entered the system and become available to determine the output. Different filter types optimize different characteristics of the ideal filter that they are meant to approximate.

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 (for example. 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 pCLAMP software 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 should have as steep a roll off as possible so that unwanted frequencies are maximally attenuated. 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 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 the pCLAMP software are 8-pole realizations.

## End Effects

All software filters exhibit end effects at the beginning or end of a data record. For 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. At the beginning of the record only succeeding points are available. These filters are 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. For the Gaussian, the filter length is equal to the number of coefficients, while for 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 $T_r = 0.35/fc$. As the filter settles to within about 10% of its final value in one rise time, a duration of 3 x $T_r$ 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.

## Finite vs. Infinite Impulse Response Filters

Digital filters can be 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 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 the pCLAMP software, the $M$ points ahead of the current point are 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 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. 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. 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:** 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 more efficient, and therefore 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. All IIR filters introduce a phase delay to the filtered data.

The problem of instability of IIR filters is solved in the pCLAMP software by limiting the cutoff frequencies for all filter types to a range where the response is always be stable. The phase delay is not corrected. See Cutoff Frequency Limitations on page 205.

The Nyquist rate has important consequences for digital filtering in that the maximum analog frequency that a digital system can represent is given by. See The Sampling Theorem in pCLAMP Software on page 23:

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

where $T$ is the minimum sampling interval and $f_h$ is the Nyquist frequency.

The maximum filter cutoff frequency of digital filters is limited to one-half the sampling rate. The ratio of the cutoff frequency to the sampling rate ($f_c/f_s$) cannot exceed 0.5. 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 lower than this because of performance degradation at higher $f_c/f_s$ ratios (see Cutoff Frequency Limitations on page 205.)

The $f_c/f_s$ ratio limitation should not present a problem if antialias filtering and oversampling are 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 digitized record.

## 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), that exhibits 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, fs = 10 kHz

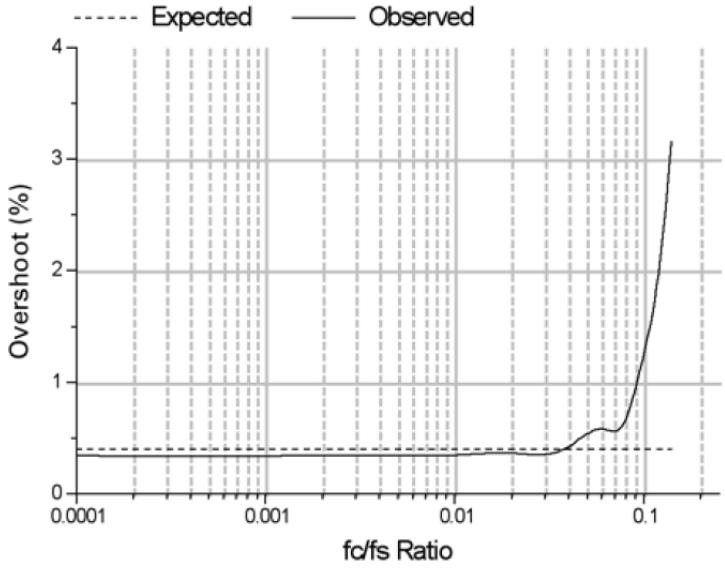



**Figure 10-1: Bessel lowpass filter (8 pole) expected vs. observed overshoot.**

### Expected vs. Observed Rise Times

100 mV step pulse, fs = 10 kHz

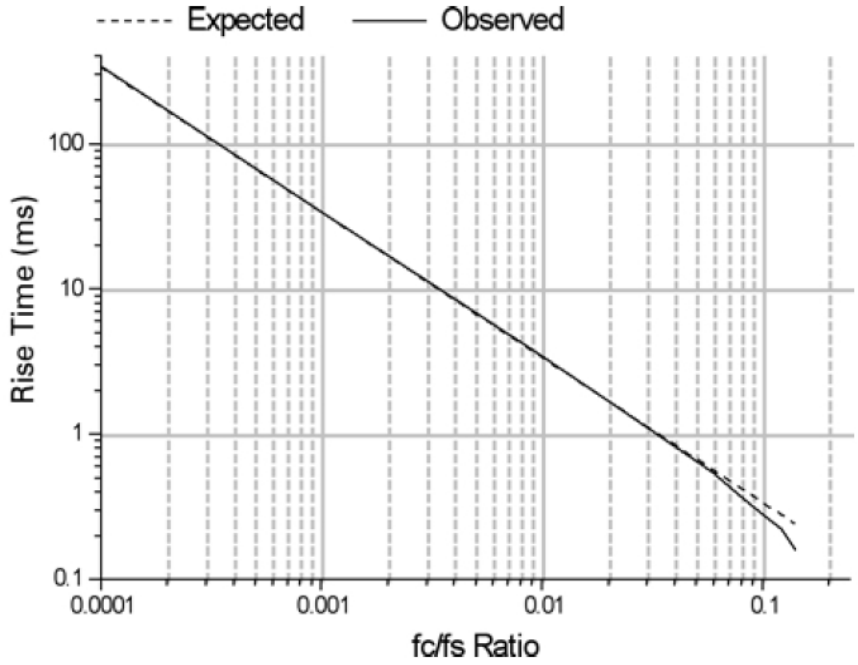



Figure 10-2: Bessel lowpass filter (8 pole) expected vs. observed rise time.

### Normalized Frequency Response

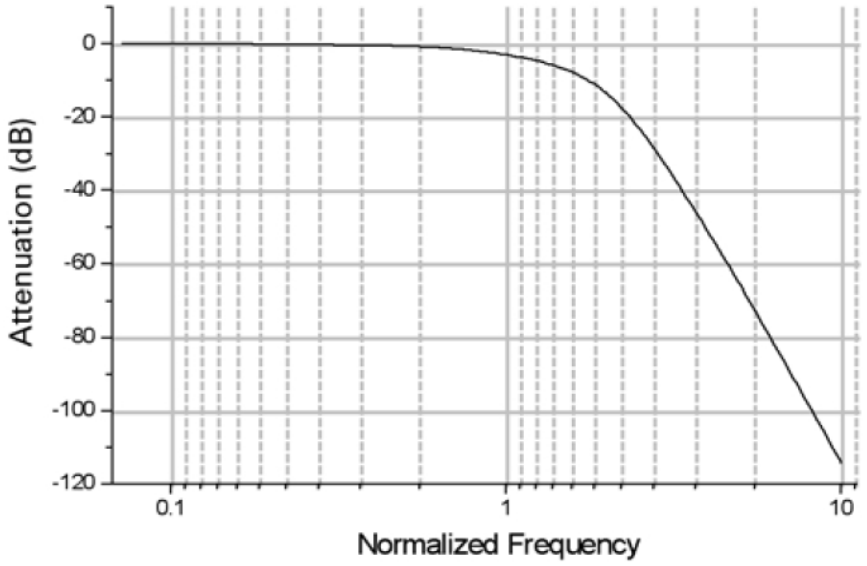



Figure 10-3: Bessel lowpass filter (8 pole) normalized frequency response.

## Boxcar Smoothing Filter Specifications

Use smoothing filters 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.

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 10-4 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

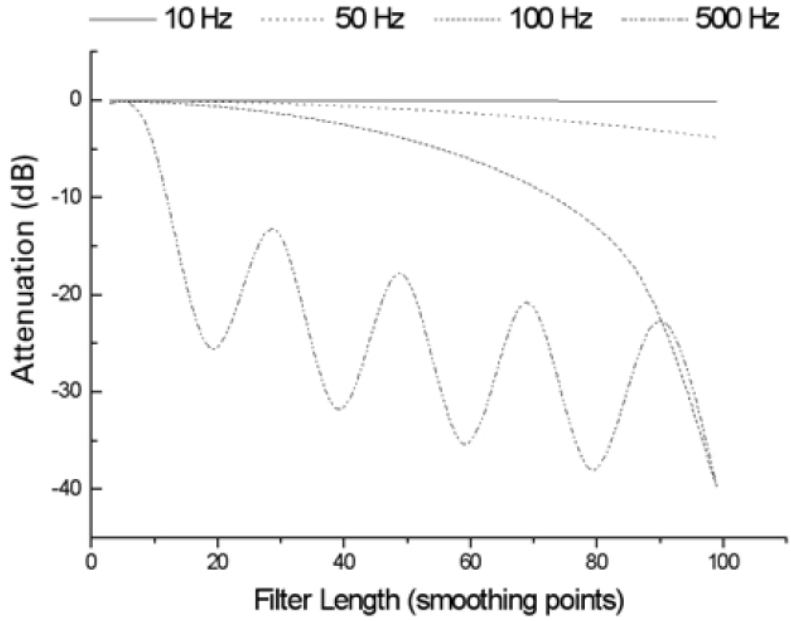



**Figure 10-4: Boxcar filter attenuation vs. number of smoothing points.**

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 has sharper roll-off characteristics than the Bessel filter. It is better suited than the Bessel for frequency domain applications such as noise analysis. Because of its nonconstant group delay characteristics, this filter should not be used for time-domain analysis of biological data.

### Expected vs. Observed Overshoot

100 mV step pulse, $f_s = 10$ kHz

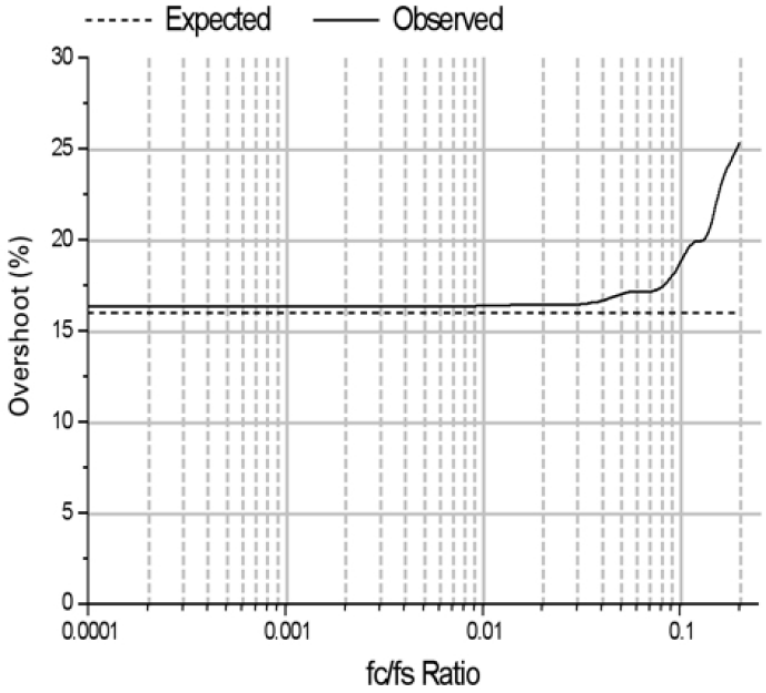



**Figure 10-5: Butterworth lowpass filter (8 pole) expected vs. observed overshoot.**

### Expected vs. Observed Rise Times

100 mV step pulse, $f_s$ = 10 kHz

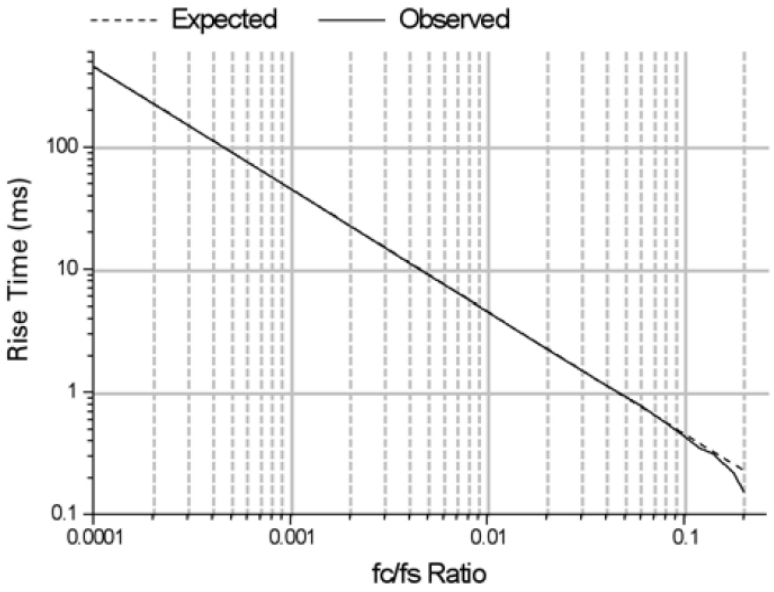



Figure 10-6: Butterworth lowpass filter (8 pole) expected vs. observed rise time.

### Normalized Frequency Response

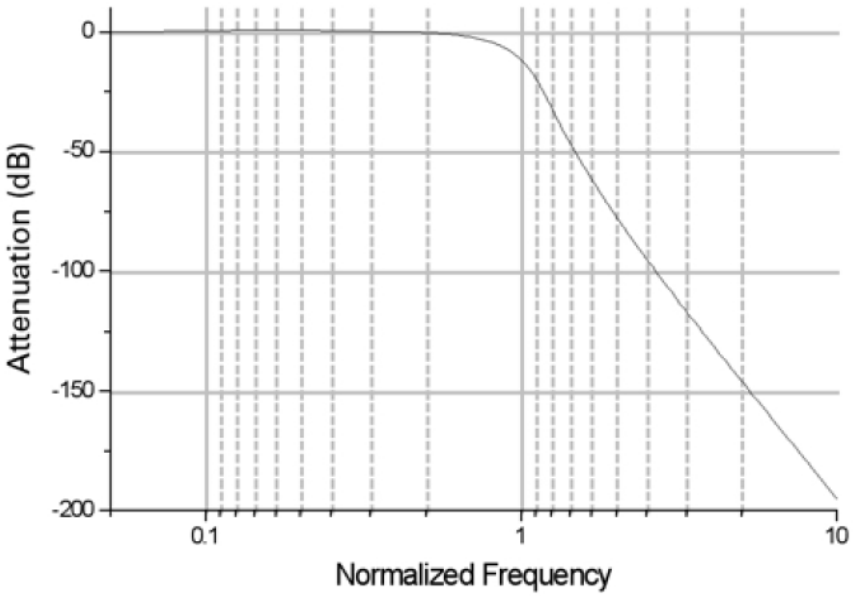



Figure 10-7: Butterworth lowpass filter (8 pole) normalized frequency response.

## 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 the pCLAMP software 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. It is not suitable for time-domain analysis of biological data.

### Expected vs. Observed Overshoot

100 mV step pulse, fs = 10 kHz

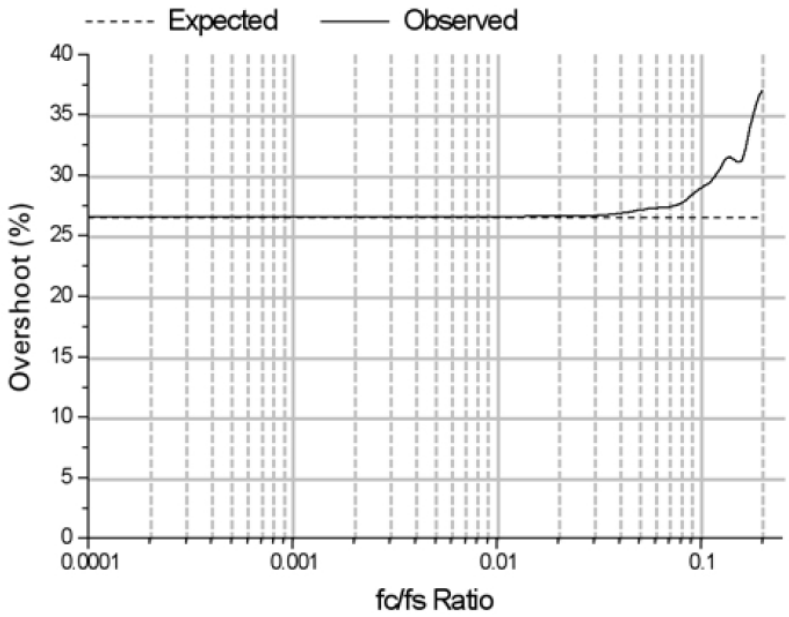



**Figure 10-8: Chebyshev lowpass filter (8 pole) expected vs. observed overshoot.**

### Expected vs. Observed Rise Times

100 mV step pulse, fs = 10 kHz

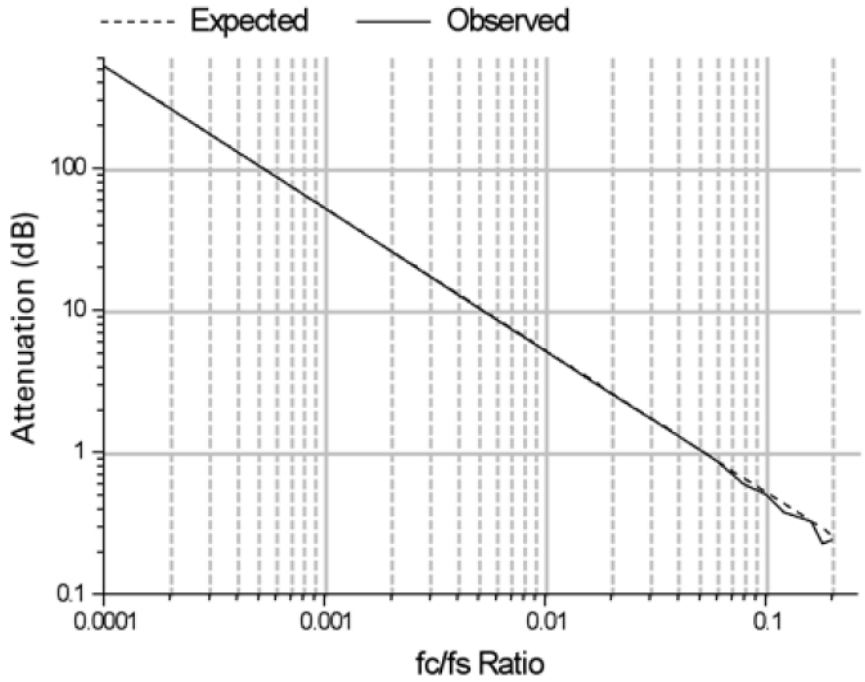



Figure 10-9: Chebyshev lowpass filter (8 pole) expected vs. observed rise time.

### Normalized Frequency Response

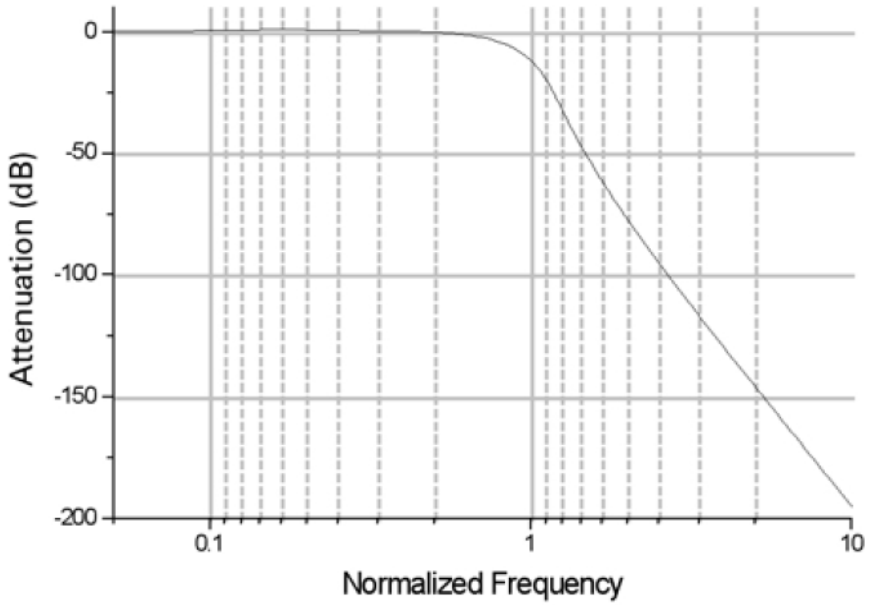



Figure 10-10: Chebyshev lowpass filter (8 pole) normalized frequency response.

## 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 described by D. Colquhoun and F.J. Sigworth (1995).

The Gaussian filter is suited to filter 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 Overshoot

100 mV step pulse, $f_s$ = 10 kHz

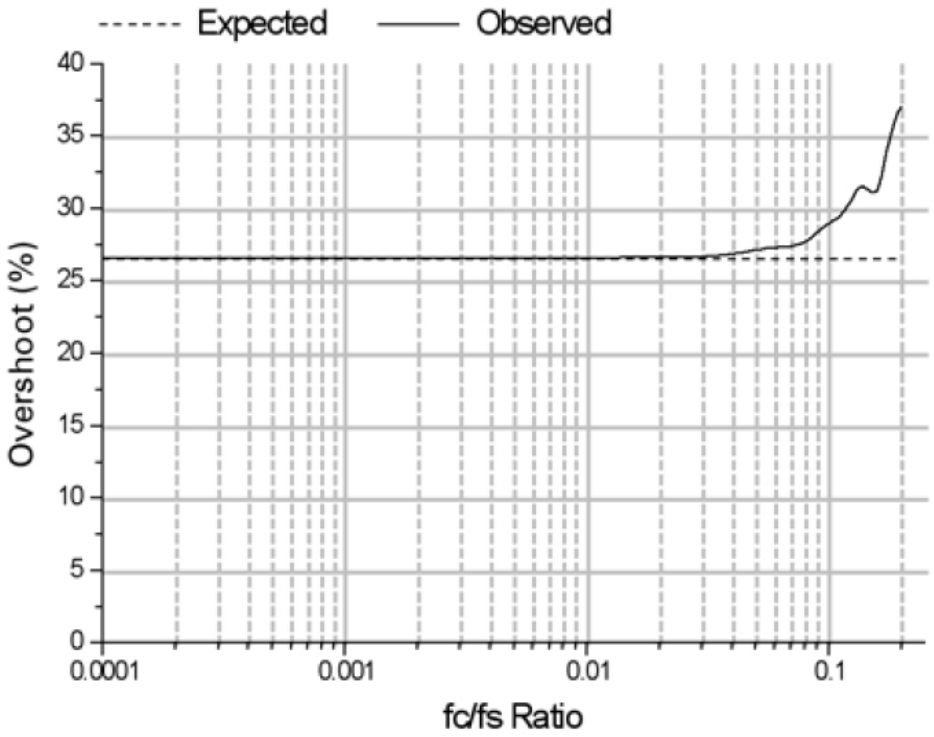



**Figure 10-11: Chebyshev lowpass filter (8 pole) expected vs. observed overshoot.**

### Expected vs. Observed Rise Times

100 mV step pulse, $f_s$ = 10 kHz

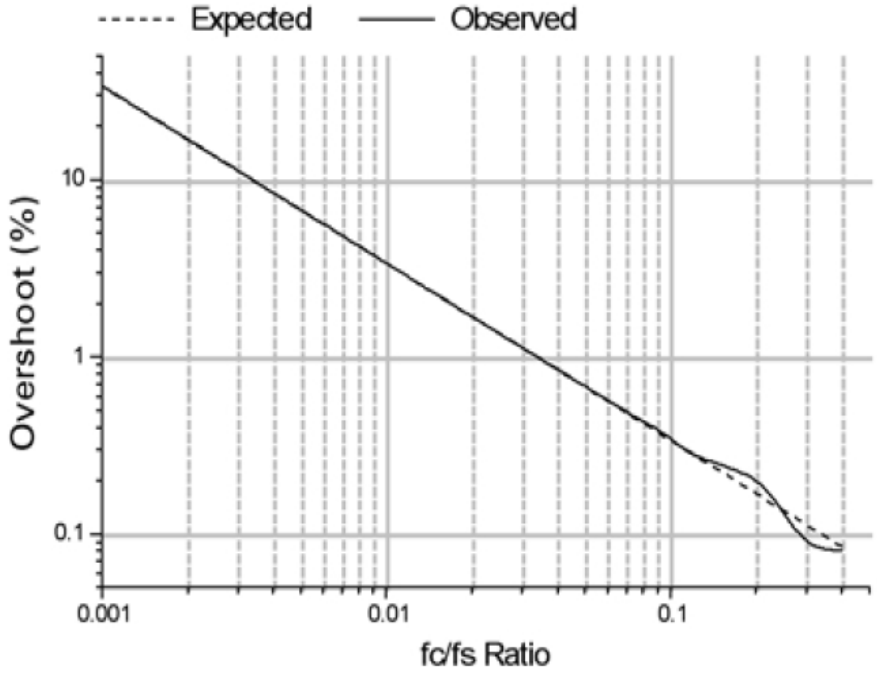



Figure 10-12: Gaussian lowpass filter expected vs. observed rise time

### Normalized Frequency Response

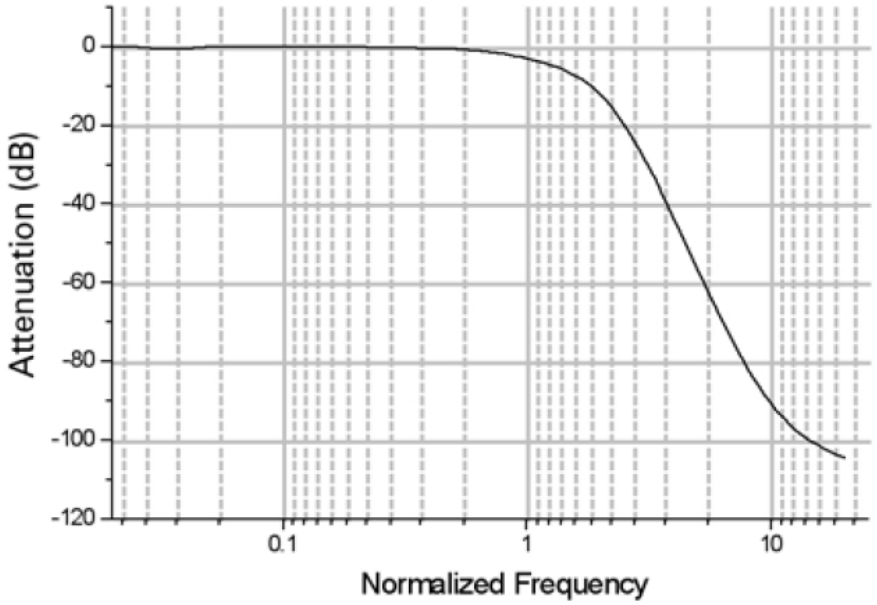



Figure 10-13: Gaussian lowpass filter normalized frequency response.

## 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

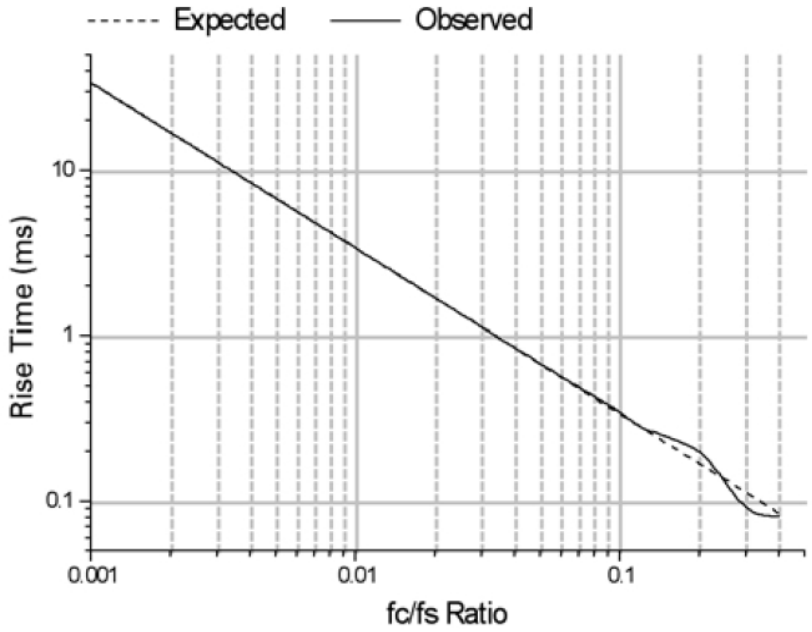



**Figure 10-14: RC lowpass filter (single pole) expected vs. observed rise time.**

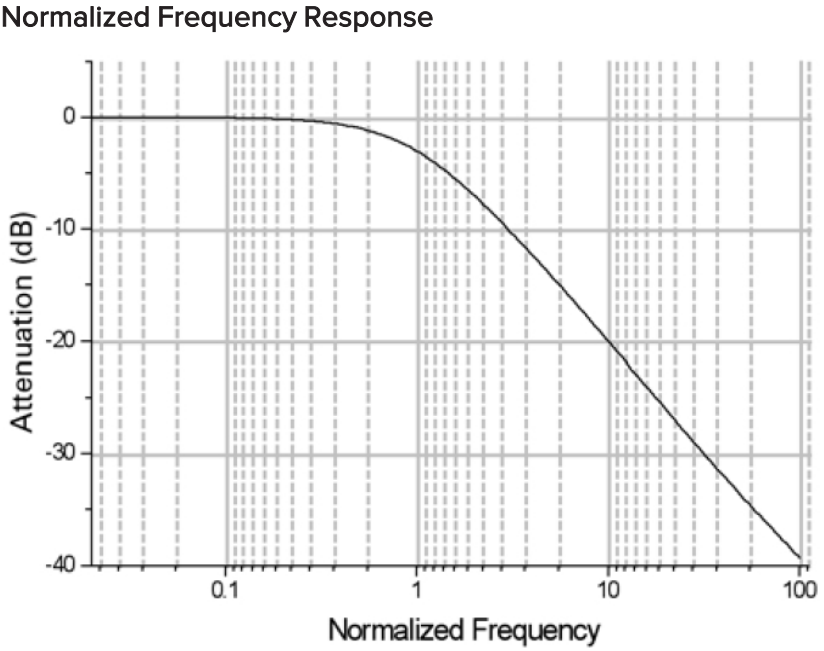



**Figure 10-15: RC lowpass filter (single pole) normalized frequency response.**

## 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 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:

$$Y(n) = X(n_p) + W + (Y(n_p - 1) - X(n_p - 1))$$

or 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:

where *n* is the order (number of poles) of the filter. For an 8-pole filter this value is 3.32397. The 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

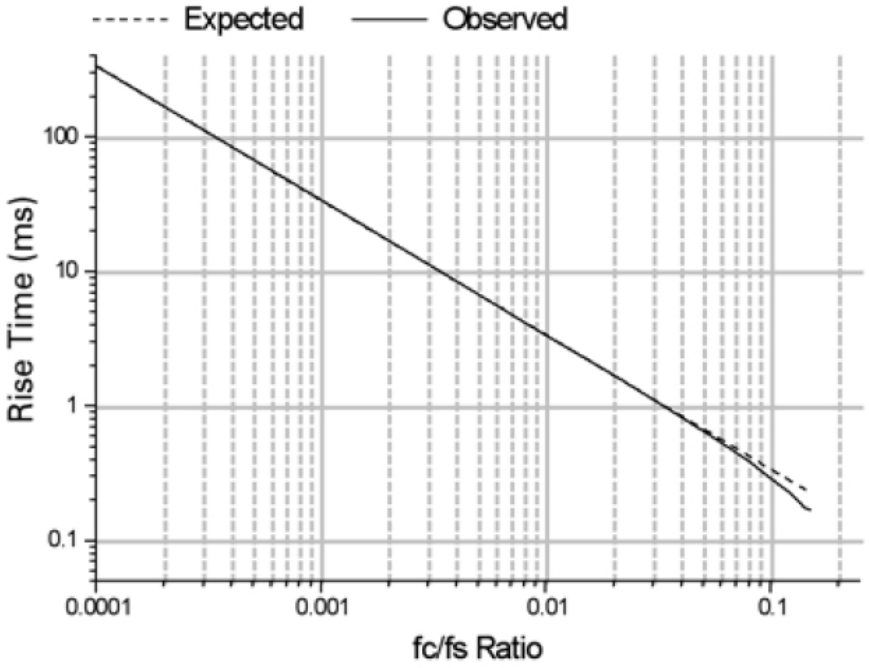



Figure 10-16: RC lowpass filter (8 pole) expected vs. observed rise time.

### Normalized Frequency Response

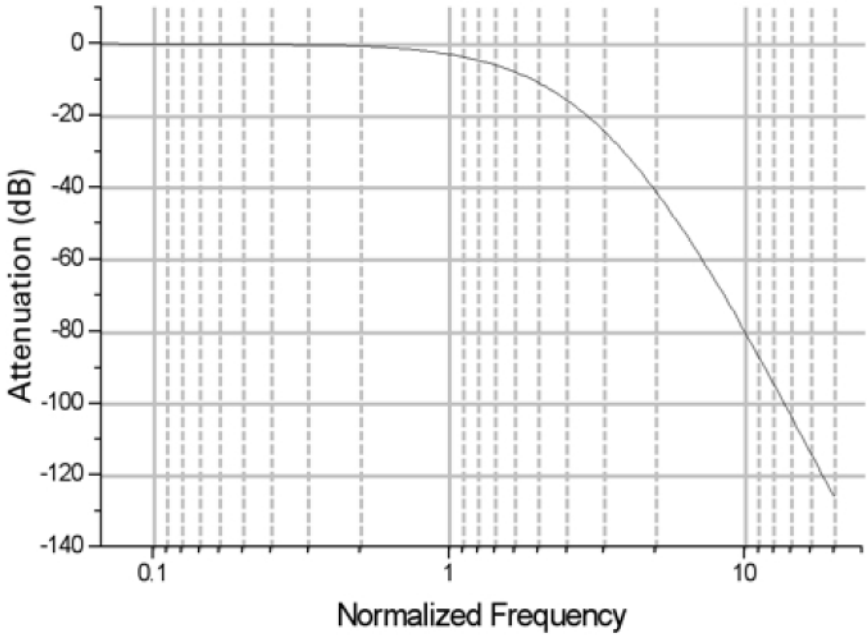



Figure 10-17: RC lowpass filter (8 pole) normalized frequency response.

## 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

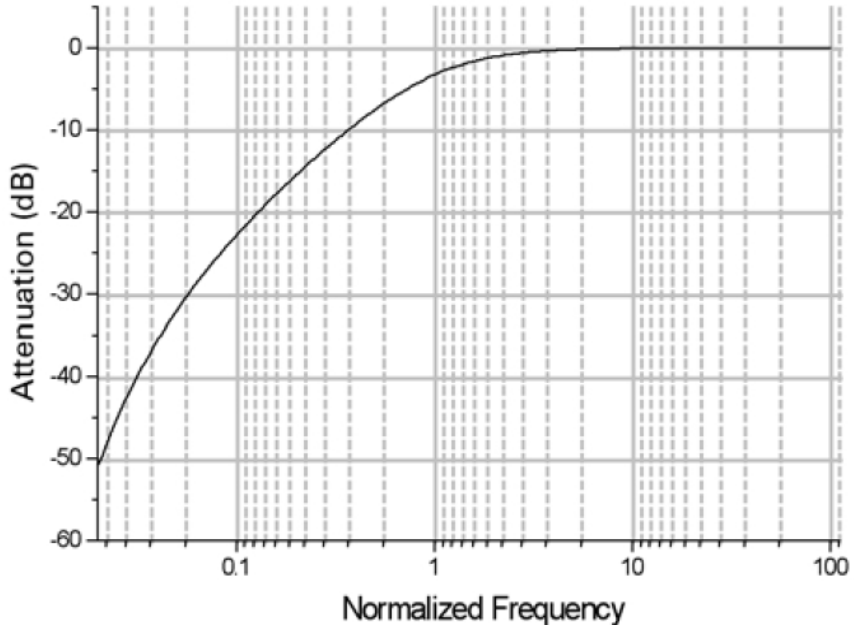



**Figure 10-18: RC highpass filter (single pole) normalized frequency response.**

## 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. 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

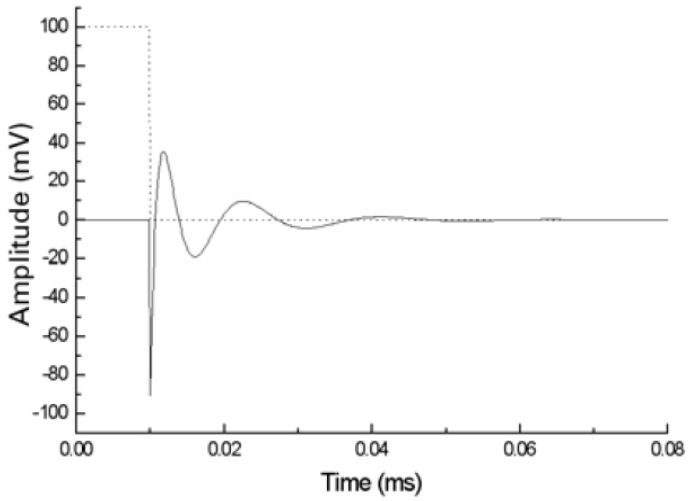



Figure 10-19: Highpass Bessel filter step response.

### Normalized Frequency Response

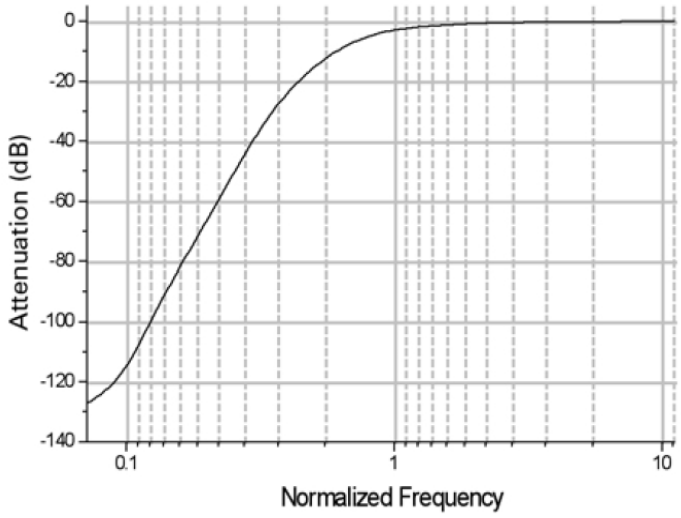



Figure 10-20: Highpass Bessel normalized frequency response.

## 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 10-21 shows the frequency response of the notch filter with a 60 Hz center frequency with a 10 Hz –3 dB width.

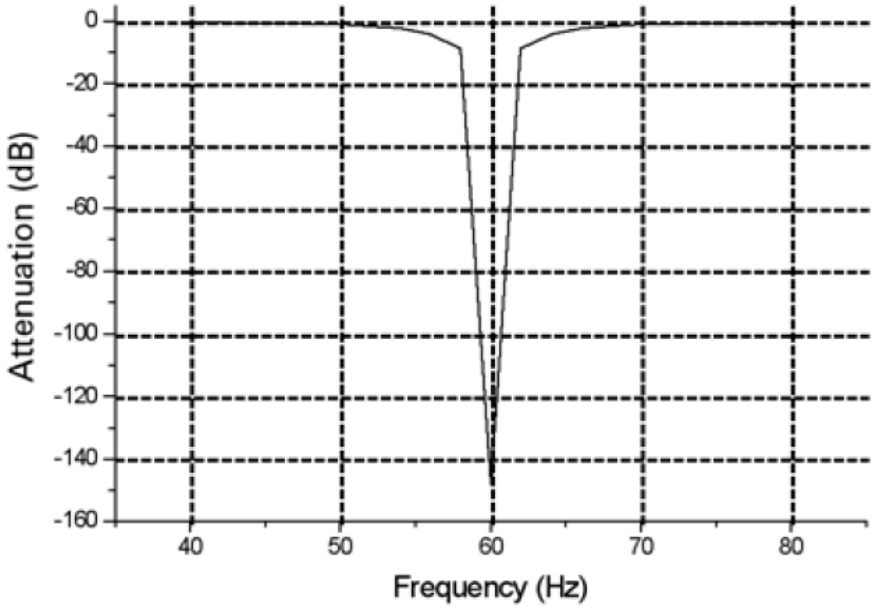



**Figure 10-21: Notch filter (2 pole) frequency response (60 Hz center frequency, 10 Hz –3 dB width)**

### Settling Points

The –3 dB width has an 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.

## The Electrical Interference Filter

The electrical interference (EI) Filter removes power line interference from a data signal. This filter identifies and removes complex power line waveforms composed of multiple harmonics. The interference detection method is adaptive. The filter always matches the actual line signal even if its properties 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 line-interference harmonic. The EI filter uses an array of sine detectors, one for each harmonic of the interference waveform.

The sine detector 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 cancelling out the complete interference waveform.

### Assumptions

- The line interference and the data signal are statistically independent.
- The line signal x(n) is stationary across at least M of its periods. You can assume that the line signal does not significantly change when observed at any M of its periods.
- 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

To identify line signal x(n) incorporated inside s(n), you assume that line signal is composed of a certain number of sinusoids with harmonic frequencies. Therefore you need to determine A and d for each harmonic.

### Basic Theory

To detect the line signal this uses the fact that data signal y(n) and line signal x(n) are uncorrelated. Practically, 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. To keep this simple, 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, this concludes that if you correlate the line signal and measured signal, you obtain the auto-correlation of the line signal. The auto-correlation function $R_{xx}(d)$ is even and its maximum is at d = 0. Since x(n) is periodic, $R_{xx}$ is also periodic with the same period.

This uses the above argument to determine the delay d and the scale A of the line signal inside the measured signal. The discussion holds for this situation if you assume that x(n) is a sinusoidal reference signal.

## Block Diagram

The adaptive line interference filter block diagram for the fundamental harmonic is shown in Figure 10-22. The same procedure is repeated for each harmonic by multiplying the frequency of the reference generator.

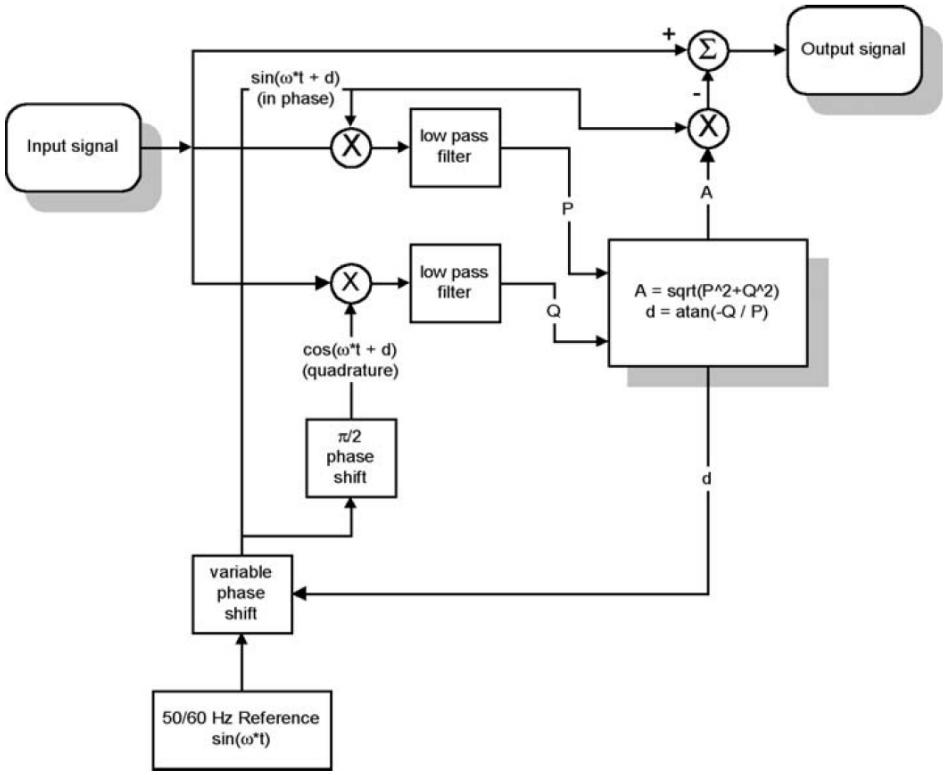



**Figure 10-22: Adaptive line interference filter block diagram.**

## 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, you reduce the correlation estimate error. Since you 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 cancelling 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 results 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 you should reduce the harmonic search number in order to exclude weak harmonics. Increasing the number of cycles to average improves harmonic detection (if the noise is the main error source). Excessively large number of cycles to average negatively affects execution speed, tracking performance, and start-up transient compensation.

## Data Signal Components

### Periodic

Any periodic components in the data signal with frequencies at or very close to 50/60 Hz multiples leaks through the EI filter harmonic detectors and produces false line interference harmonics.

Figure 10-23 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.

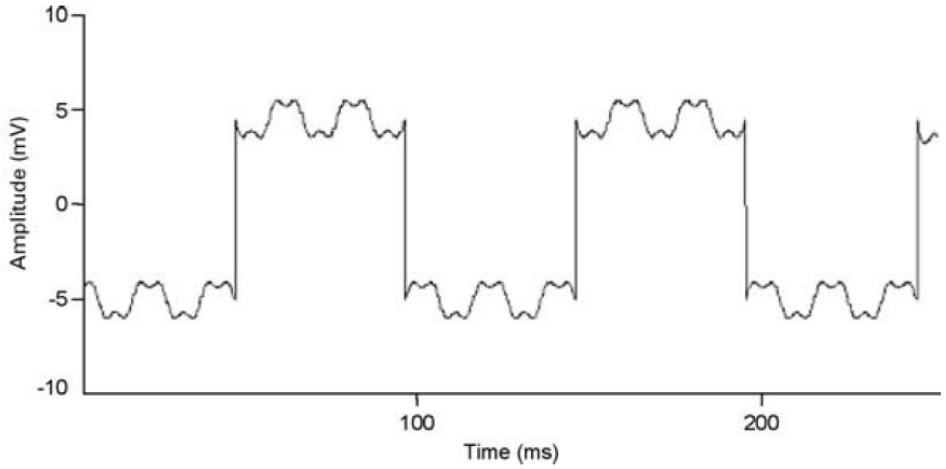



**Figure 10-23: Results of filtering a square wave.**

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 produces 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 10-24, 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.

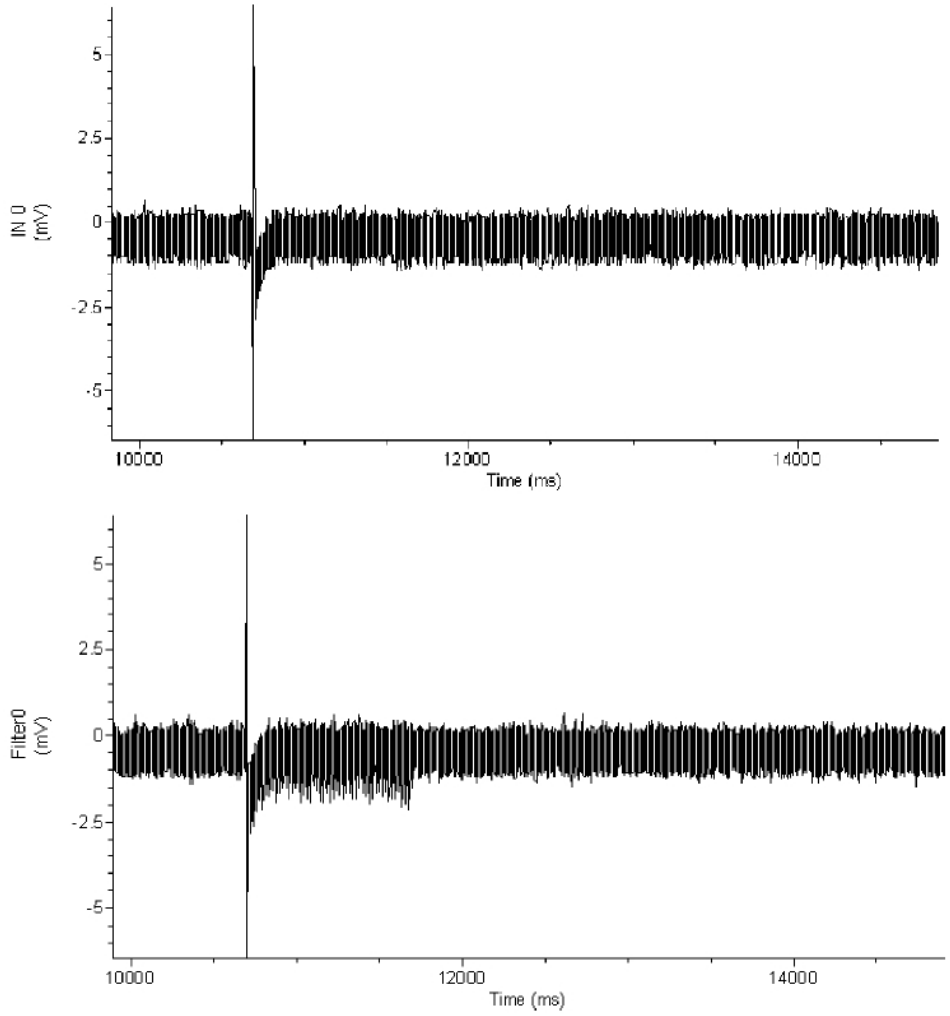



**Figure 10-24: EI filtering showing introduction of false line harmonics.**

## 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 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 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 the pCLAMP software are restricted to an $f_c/f_s$ ratio where the filter response is reasonably close to the 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. 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 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 set by the filter algorithm and cannot be altered.

**Table 10-1: 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 fs |
| Boxcar (FIR) | See note 1. | n/a |
| Butterworth (8-pole IIR) | $10^{-4}$ x $f_s$ | 0.2 x fs |
| Chebyshev (8-pole IIR) | $10^{-4}$ x $f_s$ | 0.2 x fs |
| Electrical Interference | See note 2. | n/a |
| Gaussian (FIR) | $10^{-4}$ x $f_s$. See note 3. | 0.5 x fs |
| Notch (2-pole IIR) | $10^{-3}$ x $f_s$ | 0.3 x fs |
| RC (single-pole IIR) | $10^{-4}$ x $f_s$ | 0.5 x fs |
| RC (8-pole IIR) | $10^{-4}$ x $f_s$ | 0.15 x fs. 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. 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 201). There is a data point minimum, as this filter requires a specific number of points to reach steady state. This minimum is given by:

*minimum points = samples per period x 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.

**Note: 3** The Gaussian filter width (see Finite vs. Infinite Impulse Response Filters on page 185) depends on the fc/fs ratio; the lower this ratio the greater the number of Gaussian coefficients (see Gaussian Lowpass Filter Specifications on page 193). 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 x$10^{-4}$, 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 196). 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.

# Chapter 11: Curve Fitting


The pCLAMP software offers a powerful and flexible tool to fit curves to data. Four different search methods support a 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 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 used when linear regression or polynomial regression is selected from the predefined function list. 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.

For custom functions, you must 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 be compromised.

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 123 of Clampfit Analysis Software Tutorials on page 117.

### 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 event the message *Could not compute parameter errors.* displays in the Results window.

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. 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.

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 the pCLAMP software are very robust and perform satisfactorily in most cases. There is 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, for example 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. Try a different number of terms or run a model comparison.
- Assign more accurate seeds to the fitting function. Graphical seeding should be helpful for this.
- Use the variable metric search method. This search method is the most reliable to force 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 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. 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 is 6.
- The maximum power 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 the pCLAMP software 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 tracking or assignment of units virtually impossible. Consequently, units do not display along with the reported parameter values. It is up to you to determine the units of a 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 $10^{-7}$.

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. 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 leads to a better fit, so the iterations stops.

## 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. 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. The statistics of the fit should be 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. 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:

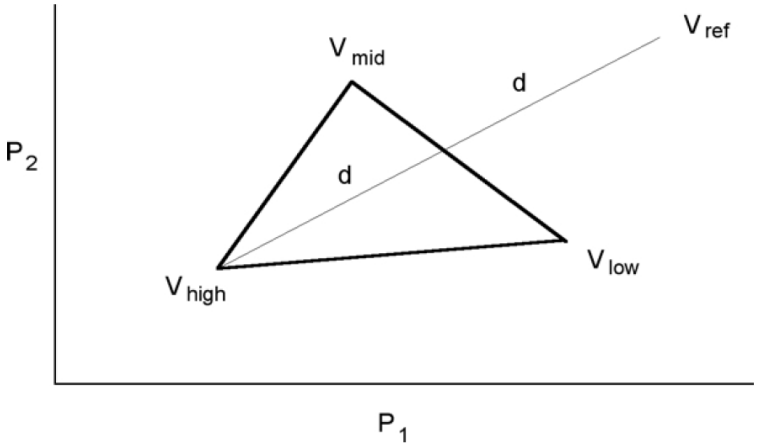



**Figure 11-1: A simplex for a two-parameter function.**

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 Vhigh. 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 rapidly approaches 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. 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.

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.

The pCLAMP software 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.

The pCLAMP software 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 231), does not change by more than a preset value over at least four iterations. This value can be set in the Precision field on the Function > Method tab in 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 following explanation 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.

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. 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 to fit 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 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 (for example $x \in [-1, 1]$), 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 of the set members. Depending on the function, a scaling factor is selected for each set member. 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 jth member of the Chebyshev polynomial set and dj is its scaling factor. A sum of an infinite number of Chebyshev polynomials is required to represent a continuous function. For a sampled dataset with N sampled points, a sum of only *N* Chebyshev polynomials, from order 0 to *N*–1, is required to represent the dataset. In this case, *t* is not continuous, but is a set of data points $t_j$, where j runs from 0 to *N*–1. The equation then becomes:

$$f(t_i) = \sum_{j=0}^{N-1} d_j T_j(t_i). \qquad \text{for } i = 0,\ldots, N-1$$

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 represented this way is said to have been transformed to the Chebyshev domain, and the scaling factors (the $d_j$s) are referred to as the coefficients of the transform. Do not confuse the transform coefficients with the 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 $d_j$s 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 ti eliminates all $T_j$s 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 $d_k$:

$$d_k = \sum_{i=0}^{N-1} T_k(t_i) f(t_i) / R_k$$

## 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 zero$^{th}$ member of this set is defined to be: $T_0 = 1$, from which all highe-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 case of a function that contains an offset; you 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 you 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 you 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, you can isolate the effect of the constant offset by not using $d_0$ in the calculations of the other parameters.

## 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 of the set members. Depending on the function, a scaling factor is selected for each set member. 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 jth member of the Chebyshev polynomial set and dj is its scaling factor. A sum of an infinite number of Chebyshev polynomials is required to represent a continuous function. For a sampled dataset with N sampled points, a sum of only *N* Chebyshev polynomials, from order 0 to *N*–1, is required to represent the dataset. In this case, *t* is not continuous, but is a set of data points $t_j$, where j runs from 0 to *N*–1. The equation then becomes:

$$f(t_i) = \sum_{j=0}^{N-1} d_j T_j(t_i). \qquad \text{for } i = 0,\ldots, N-1$$

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 represented this way is said to have been transformed to the Chebyshev domain, and the scaling factors (the $d_j$s) are referred to as the coefficients of the transform. Do not confuse the transform coefficients with the 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)*.

## Integrating an Exponential Function in the Chebyshev Domain

When you have a function *f(t)* that you want to integrate, and that its integral is *F(t)*. In the discrete domain, where you only have a finite set of data points (usually evenly spaced), you 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 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 you transform f to the Chebyshev domain, you would obtain a set of coefficients $d_j(f)$. Similarly, if you transform *F*, you would obtain a different set of coefficients $d_j(F)$. Comparing these two sets of coefficients, $d_j(f)$ and $d_j(F)$ you would find the relation:

$$D_j(F) = \frac{1}{2}\left(\frac{(N+j+1)}{2j+3}d_{j+1}(f) - d_j(f) - \frac{N-j}{2j-1}d_{j-1}(f)\right) \quad \text{for all } j > 1$$

where $d_j(F)$ is the $j^{th}$ coefficient of $F(t)$ and $N$ is the number of points in the data. This equation cannot tell you 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. 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:

$$\frac{d_j(f)}{k} = \frac{1}{2}\left(\frac{(N+j+1)}{2j+3}d_{j+1}(f) - d_j(f) - \frac{N-j}{2j-1}d_{j-1}(f)\right) \quad \text{for all } j > 1$$

where $k$ is defined as:

$$k = e^{-1/\tau} - 1$$

or, solving for τ:

$$\tau = \frac{-1}{log_e(k+1)}$$

Basically, these equations tell you that any adjacent triplet of Chebyshev coefficients forms an exact relationship that tells you 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.

Integrating an exponential function in the Chebyshev domain allows you 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:

$$\frac{d_j(f)}{k} = d1_j(f) \qquad \text{for } j > 1$$

where $k$ is as defined in Equation 5.

## Calculating Tau

Now you 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, you must change directions and generate an exponential dataset based on the value of tau just calculated. The generated dataset has unity amplitude and zero offset:

$$g(t_i) = 1e^{-t_i/\tau} + 0$$

Transforming this generated set to the Chebyshev domain gives you a different set of coefficients ($d'_j(g)$). Now you can determine the value of the amplitude of the 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 you are trying to fit is:

$$f(t_i) = a_0 + a_1 e^{-t_i/\tau}$$

you 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 similar to those for *f(ti)*:

$$d'_j(g) = \sum_{i=0}^{N} e^{-t_i/\tau} \cdot T_j(t_i)/R_j \qquad \text{for all } j$$

Comparing these two equations yields the following relationship between the Chebyshev coefficients of *f* and those of *g*:

$$a_1 = \frac{d_j(f)}{d'_j(g)} \qquad \text{for all } j > 0$$

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: you compare the zeroth index coefficients from the two sets of transforms:

$$a_0 = d_0(f) - a_1 d'_0(g)$$

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; $d_i(f)/k$ is an estimate of the right side of this equation. Equation 6 then becomes:

$$d_j(f) \cong k d1_j(f)$$

The value of *k* that minimizes the following expression is *k*′:

$$\chi^2 = \sum_{j=1}^{n} (d_j - k d1_j^2(f))$$

where the sum does not include all *N* of the coefficients, but 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 you an estimate of *k*:

$$k' = \sum_{j=1}^{n} d_j d1_j(f) / \sum_{j=1}^{n} d1_j^2(f)$$

Once *k* has been estimated, Equation 5 gives the corresponding best estimate of τ.

## Estimating $a_1$

Use a similar technique to calculate the best estimate of the amplitude $a_1$. The ratios of the coefficients of the 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, you 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 you:

$$a'_1 = \sum_{j=1}^{n} d_j(f) d'_j(g) \Big/ \sum_{j=1}^{n} (d'_j(g))^2$$

## 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, you integrate the function to be fit twice, and solve the resulting set of simultaneous equations for $\tau_1$ and $\tau_2$. (This procedure is somewhat complicated.) Once you have the two taus, solving for the amplitudes and the offset is a simple extension of the procedure for fitting a single exponential.

Suppose you want 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 $\tau_1$ and let $g_{\tau 2}(t_i)$ represent a unity-amplitude exponential with time constant $\tau_2$. Then you can write:

$$f(t_i) = a_0 + a_1 g_{\tau_1}(t_i) + a_2 g_{\tau_2}(t_i)$$

Transforming both sides of this equation gives you:

$$d_j(f) = \left( \sum_{i=0}^{n} (a_0 + a_1 g_{\tau_1}(t_i) + a_2 g_{\tau_2}(t_i)) T_j(t_i) \right) / R_j$$

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 $d_j^{offset}(a_0)$, $dj^{\tau 1}(g_{\tau 1})$, and $d_j^{\tau 2}(g_{\tau 2})$ are the Chebyshev coefficients of $a_0$, $g_{\tau 1}$ and $g_{\tau 2}$ respectively. To isolate the constant term from the calculations that follow, you only use coefficients where $j > 0$, yielding:

$$d_j(f) = a_1 d_j^{\tau_1}(g_{\tau_1}) + a_2 d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 0$$

since the offset is only contained in the zero$^{th}$ coefficients.

## Using the Coefficients of f and its Integrals to Determine the Taus

What if you were to integrate both sides of Equation 12? Since you are dealing with discrete data points, you use a summation. From Equations 4, 5 and 6, you know the coefficients of the integral of an exponential function, and you 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:

$$d1_j(f) = a_1 d1_j^{\tau_1}(g_{\tau_1}) + a_2 d1_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 1$$

or using Equation 6 to rewrite in terms of the coefficients of and themselves:

$$d1_j(f) = \frac{a_1}{k_1} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2} d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 1$$

where

$$\tau_1 = \frac{-1}{log_e(k_1 + 1)} \quad \text{and} \quad \tau_2 = \frac{-1}{log_e(k_2 + 1)}$$

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 you write the Chebyshev coefficients of the second integral of $f$ as $d2_i(f)$. $j$ now must be greater than two. (The exact reason for this is beyond the scope of this description (see Malachowski's paper) however, it is required to isolate the effect of the offset from the calculation of the taus.) Substituting in Equation 6 again gives you the final relation that you need to determine $\tau_1$ and $\tau_2$:

$$d2_j(f) = \frac{a_1}{k_1^2} d_j^{\tau_1}(g_{\tau_1}) + \frac{a_2}{k_2^2} d_j^{\tau_2}(g_{\tau_2}) \qquad \text{for } j > 2$$

## Solving a Set of Simultaneous Equations to Determine $k_1$ and $k_2$

Equations 12, 14 and 16 now form the three relations that you need in order to determine the taus. Rewriting them below you 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 $j > 2$:

$$d_j + x_1 d1_j + x_2 d2_j = 0$$

The solution to this equation is a straight line in $x_1$–$x_2$ coordinate space. To solve it, you 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:

$$x_1 = -(k_1 + k_2)$$

$$x_2 = k_1 k_2$$

as can be seen by substituting these values into the above equation.

The strategy is 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 $\tau_1$ and $\tau_2$. To do so, you 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 you can recognize that Equations 18a and b are the roots of the quadratic polynomial:

$$k^2 + x_1 k + x_2 = 0$$

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 you 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, you 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 yields 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. It is the product of an exponential and a harmonic function (for example, a cosine). This function is 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 $\theta$ 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 $d2_j$ 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, you form the following regression equation, and minimize the $\chi^2$ value:

$$\chi^2 = \sum_{j=2}^{n} (d_j - x_1 d1_j - x_2 d2_j)^2$$

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 you have the following set of simultaneous equations:

$$\sum_{j=2}^{n} d_j d1_j = x_1 \sum_{j=2}^{n} d1_j^2 + x_2 \sum_{j=2}^{n} d1_j d2_j$$

$$\sum_{j=2}^{n} d_j d2_j = x_1 \sum_{j=2}^{n} d1_j d2_j + x_2 \sum_{j=2}^{n} d2_j^2$$

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 you to 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.) You generate two exponential datasets based on the values of $\tau_1$ and $\tau_2$ 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, 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:

$$\chi^2 = \sum_{j=1}^{n} \left( d_j(f) - a_1 d_j^{\tau_1}(g_{\tau_1}) - a_2 d_j^{\tau_2}(g_{\tau_2}) \right)^2$$

Solution of this equation is not shown, but involves expanding the square inside the summation, minimizing first with respect to $a_1$ and $a_2$, 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:

$$a_0 = d_0(f) - a_1 d_0^{\tau_1}(g_{\tau_1}) - a_2 d_0^{\tau_2}(g_{\tau_2})$$

## 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 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.

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 (for example 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: for example the single channel events may still be seen). This occurs since low-frequency noise appears 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.

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 that have 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:** The Chebyshev search method had not been published in a peer-reviewed journal. While many empirical tests that have confirmed the accuracy of the fits have been conducted, you should 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 the fitting parameters. Exponentially distributed data ($t_i$) are described by components ($k$) of the form:

$$F(t_j) = \sum_{j=1}^{k} a_j e^{-t_i/\tau_j}$$

where $t_1, t_2, \ldots t_n$ are the n measured data points and $\tau_j$ 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 $t_i$:

$$f(t_j) = \frac{d}{dt_i} F(t_j) = \sum_{j=1}^{k} a_j \tau_j e^{-t_i/\tau_j}$$

The likelihood ($L$) of obtaining a set of observed data $t_i$, given the form of the distribution and the set of function parameters ($\theta$) is the product of the probabilities of making each of the $N$ observations:

$$L = \prod_{j=1}^{N} f(t_j|\theta)$$

As the likelihood takes on small values the numerical evaluation of its logarithm is preferable. Limited frequency resolution of the recording system makes it impossible to record events shorter than $t_{min}$. This case assumes that no events longer than $t_{max}$ can be measured. Thus, 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 measurable dwell-times fall within the range delimited by $t_{min}$ and $t_{max}$ given the probability distribution with parameters $\theta$. In the software $t_{min}$ and $t_{max}$ are defined by the lower and upper limits of the data time base.

This fitting algorithm is not a maximum likelihood estimation. It is iteratively re-weighted least squares fit to the number of elements in a bin. The weighting relates to the expected variance of the number of elements in a bin, which is Poisson distributed. The weighting factor is the inverse of the variance. The iterative aspect relates to the fact that the number of elements per bin get moved around to correct for censoring. Because of the simple form of the weighted sum of exponentials the derivatives of the minimized function with respect to each parameter can be written directly and corrected for the Poisson-distributed variance.

## Maximum Likelihood Estimation

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:

$$L_b(\theta) = \sum_{i=1}^{N} n_i \ ln\left\{\frac{F(t_{i+1}|\theta) - F(t_i|\theta)}{p(t_s,t_k|\theta)}\right\}$$

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

$$p(t_s,t_k|\theta) = F(t_s|\theta) - F(t_k|\theta)$$

In pCLAMP software, 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 $\tau_i$. The coefficients ai sum to unity, where:

$$\sum_{i=1}^{m} a_i = 1$$

Therefore, the set of parameters θ has $2m–1$ degrees of freedom. pCLAMP software can use either the Simplex or variable metric search methods to find the set of parameters that maximize $L_b(\theta)$. 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 pCLAMP software 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 (for example different orders) produce fits that are visually very similar. pCLAMP software 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 appear 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.

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 $\beta$ and $\theta$ 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 $\beta$ and $\theta$ 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 pCLAMP software by “F”.

The degrees of freedom ($k_f$ and $n–k_g$) are 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. 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 Minimization on page 236 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 is 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:

$$SSE = \sum \left[ \left( \left( f_i^{obs} - f_i(\theta) \right)^2 / f(\theta) \right) \right]$$

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 $f(\theta)$ is 1.

## Function

Weighting the sum of squared errors function generates the Chi-square ($\chi^2$) function. The ($\chi^2$) value for a function with a given set of parameters $\theta$ 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, $f_i(\theta)$ 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 $\chi^2$ 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, $f_i(\theta)$ is the fit function calculated for the $i^{th}$ data point, and fiobs 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.

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 216), 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 11-2 shows a fit of the Z-delta Boltzmann function to current/voltage data. The fitted curve is the line without symbols.

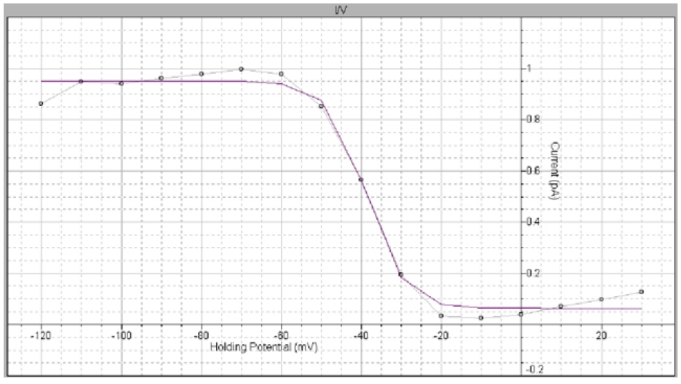



**Figure 11-2: Fit of Z-delta Boltzmann function to current/voltage data.**

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.

# Chapter 12: Fitting Functions


The pCLAMP software provides predefined fitting functions to assist in analysis. Some functions require that restrictions, such as forced positive parameters, are applied during the fit. Some options in the Fit dialog might be set to comply with mandatory fitting requirements. Some functions require automatic data preprocessing, such as normalization, prior to fitting. In the function descriptions below, any 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 details of the functions and examples of 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. Use this method 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 normalized prior to fitting. These values are rescaled after fitting so that the fitted curve conforms to the original data.

The beta function is intended to be used 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. The pCLAMP software 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, use this function to determine the probability for a 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 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 appears as a histogram.

The recommended fitting method is Levenberg-Marquardt.

## Boltzmann, Charge-Voltage

$$f(V) = \frac{I_{max}}{1 + e^{(V_{mid} - V)/V_c}} + C$$

Use this function 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, $\tau$ is the slope and $C$ is a constant offset in the $y$ direction (see Boltzmann, Standard on page 244). 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 $A$, $B$, $\tau$ 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 you 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*. *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

$$f(V) = V_{min} + \frac{V_{max} - V_{min}}{1 + e^{\frac{Z_d F}{RT}\langle V - V_{mid}\rangle}}$$

Use this function 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 halfmaximal, *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 used to describe voltage-activated sodium and potassium currents (with *a* = 3 and *b* = 1). The term $k_\infty$ is the steady-state inactivation, $I'$ is the maximum current that is achieved in the absence of inactivation, $\tau_j$ is the activation time constant, $\tau_k$ is the inactivation time constant, and the power terms a and *b* are empirically determined (Dempster, 1993, pages 140–142).

The fit solves for $I'$, $\tau_j$, $\tau_k$, $k_\infty$ and the constant *y*-offset *C*. The power terms a and b must be specified.

The recommended fitting method is Levenberg-Marquardt.

## Exponential, Alpha

$$f(t) = \sum_{i=1}^{n} A_i t e^{-t/\tau_i} + C$$

Use the alpha exponential function 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. Each successive bin contains its own data plus the data in all of the preceding bins.

Do not use this function 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 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

$$f(t) = \sum_{i=1}^{n} P_i \tau_i^{-1} e^{-t/\tau_i} + C$$

Use this function 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

$$f(t) = \sum_{i=1}^{n} A_i\left(1-e^{-t/\tau_{r_i}}\right)\left(e^{-t/\tau_{d_i}}\right) + C$$

Use this function to fit postsynaptic events (excitatory or inhibitory postsynaptic potentials). The fit solves for the amplitude *A*, the rise time constant $\tau_r$ and the decay time constant $\tau_d$ for each component *i*.

The recommended fitting method is Levenberg-Marquardt.

## Exponential, Sloping Baseline

$$f(t) = \sum_{i=1}^{n} A_i e^{-t/\tau_i} + mc + C$$

Use this function 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

$$f(t) = \sum_{i=1}^{n} A_i e^{-t/\tau_i} + C$$

Use this function 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.

Use this function 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

$$f(t) = \sum_{i=1}^{n} A_i \frac{e^{-(x-\mu_i)^2/(2\sigma_i^2)}}{\sigma_i\sqrt{2\pi}} + C$$

Use this 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.

Use this function to describe 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.

Use this function 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.

Use this function to describe the steady-state dependence of membrane voltage on ion concentrations and the relative permeability of those ions through the membrane.

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, you use this equation to calculate the relative permeability of the activated conductances 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 you use to fit 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

$$f(x) = \sum_{i=1}^{n} \frac{I_{max_i}[x]^{h_i}}{C_{50_i}^{h_i} + [x]^{h_i}} + C$$

Use the Langmuir-Hill equation to fit a sum of Langmuir components. It is useful to fit 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 you can apply 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 has 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 is derived by taking the reciprocal of both sides of the Michaelis-Menten equation to describe a straight line with a slope of $K_m/V_{max}$ and a *y*-intercept of $1/V_{max}$. Use this to gain 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 increases when B is positive and decrease when $B$ is negative.

The fit solves for Rmax and $A$, $B$ and the constant $y$-offset $C$.

The recommended fitting method is Levenberg-Marquardt.

## Lorentzian Distribution

$$f(x) = \sum_{i=1}^{n} \frac{2A_i\omega}{4\pi(x-\mu)^2 + \omega^2} + C$$

Use the Lorentzian distribution function 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 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 $\tau_c$ can be ignored and $\tau$ can be equated to the channel open time constant $\tau_o$. 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 $S(0)$, $\tau$ and the constant *y*-offset $C$ for each component $i$.

The parameter $\tau$ has units of *s* if the frequency, $f$, is in Hz.

The recommended fitting method is Levenberg-Marquardt.

## Michaelis-Menten

$$f(S) = \frac{V_{max}[S]}{[S] + K_m} + C$$

Use this 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

$$V(x) = \frac{RT}{zF} ln \frac{[x]_1}{[x]_2}$$

Use this function to describe 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 you use 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 appears as a histogram.

The recommended fitting method is Levenberg-Marquardt.

## Polynomial

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

This 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 $f(x) = a_0x^0 + a_1x^1 = a_0 + a_1x$, which is a straight-line fit.

The maximum order is 6.

## Straight Line, Origin at Zero

$$f(x) = ix$$

Use this function 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

# Chapter 13: Troubleshooting

13

## Software Problems

If you have installation questions or problems, first check the Knowledge Base linked through the Help menu. We continually improve our products, and often a problem has been fixed in a newer software version. Download the latest version of the pCLAMP software and software release notes from the Knowledge Base.

You should your hard drive and file structure when you encounter problems. For problems not related to data acquisition, try more than one data file to see if the problem is due to a corrupted data file. If the problem is associated with one protocol file, that file may be corrupted; use **Acquire** > **New Protocol** to recreate the protocol. If the program generates severe errors, use the utility in the Molecular Devices folder, Reset to Program Defaults, to clear the registry entries.

**CAUTION!** If you use the Clampfit Software Advanced Analysis Batch Analysis functionality:
Do not clear the registry before noting the active database name and path in the Batch Analysis status bar. Clearing the registry resets the database connection to the default Batch Analysis database and you must use the Select Database option to reconnect to the database .

Context-sensitive menus are used. If you cannot to find or access a feature, it might be associated with a different type of window from the one that is active, for example that has its title bar highlighted. Or, an acquisition mode might need to be active for associated operations to be available. Sometimes, a certain parameter needs to be selected to enable additional parameters. Refer to the Help file.

## Hardware Problems

If you have other equipment hooked up to the 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 cannot determine if you have a hardware or a software problem, try moving the 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 hardware problems are defective cables. Try using different cables.

## Obtaining Support

Molecular Devices is a leading worldwide manufacturer and distributor of analytical instrumentation, software, and reagents. We are committed to the quality of our products and to fully supporting our customers with the highest level of technical service.

Our Support website—support.moleculardevices.com/—describes the support options offered by Molecular Devices, including service plans and professional services. It also has a link to the Molecular Devices Knowledge Base, which contains documentation, technical notes, software upgrades, safety data sheets, and other resources. If you still need assistance, you can submit a request to Molecular Devices Technical Support.

Please have your instrument serial number or Work Order number and your software version number available when you call.

# Appendix 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 Ca2+-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 patchclamp 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.

# Appendix B: Resources


## Programs and Sources

Several software programs complement the pCLAMP software by providing additional analysis and plotting functions.

**Note:** Support of the pCLAMP 11 .abf file format has not been verified with these companies.

**AxoGraph X** (AxoGraph Scientific) 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 the pCLAMP 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.

**Contact Us**

Phone: +1-800-635-5577
Web: moleculardevices.com
Email: info@moldev.com

Visit our website for a current listing of worldwide distributors.


1-2500-0180 N

MOLECULAR DEVICES