Guide

elements
enabling technologies for Life Science

EMCR - Online Analyses

ELEMENTS srl - ITALY - C.F/P.IVA/VAT 04113900403 - www.elements-ic.com

commercial info: info@elements-ic.com - technical support: support@elements-ic.com

Revision History

Date	Version	Description
11/03/2025	0.1.0	First version of document

Introduction

This guide shows more in detail the different online analyses that can be performed in EMCR, and how they work.

Currently there are two types of analysis in EMCR: Central Widget Analyses and Protocol Based Analyses

Central Widget Analyses

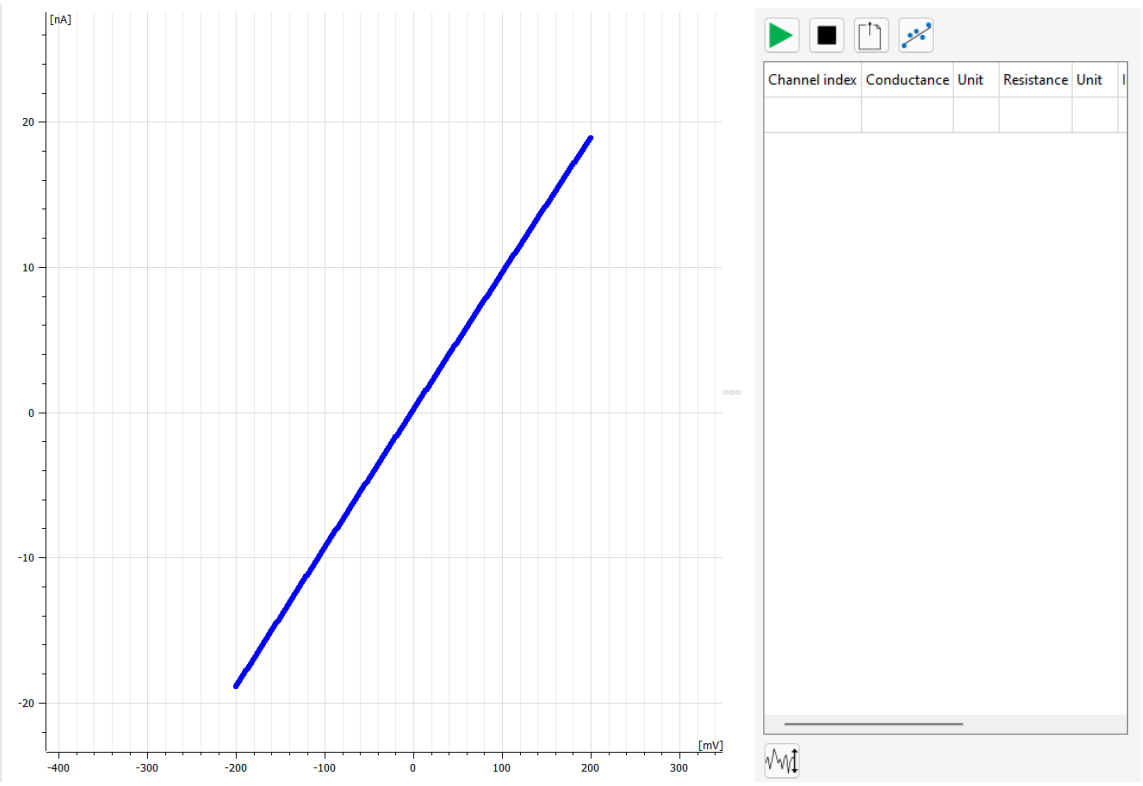
These analyses can be accessed from the tabs in the central widget. These analyses are characterized by the fact that their results have a graphical part which takes the place of the time domain plot, and that they can be applied independently of the applied stimulation protocol. The currently available central widget analyses are:

- IV Graph
- Event Detection
- Spectrum

IV Graph

The IV graph will display a plot that has currents on the y axis and voltages on the x axis.

It is important to note that this kind of analysis **only works on channels that are Expanded**.



IV Graph Controls

- Start the IV graph analysis: a reinitialization of the data will be performed to discard old data from previous protocols.
- Stop the analysis: stop the data flow from the device, the plot will not be updated anymore (until the start analysis button is pressed again)
- Export the data: export the plotted points to a CSV file of your choice
- Least square line: approximates the IV graph data with a line and populates a table with useful information. You can copy these values and paste them in a file of your choice; they will be tab separated in order to enable the user to open them as a spreadsheet
- Auto zoom the x and y axis.

IV Graph Algorithm

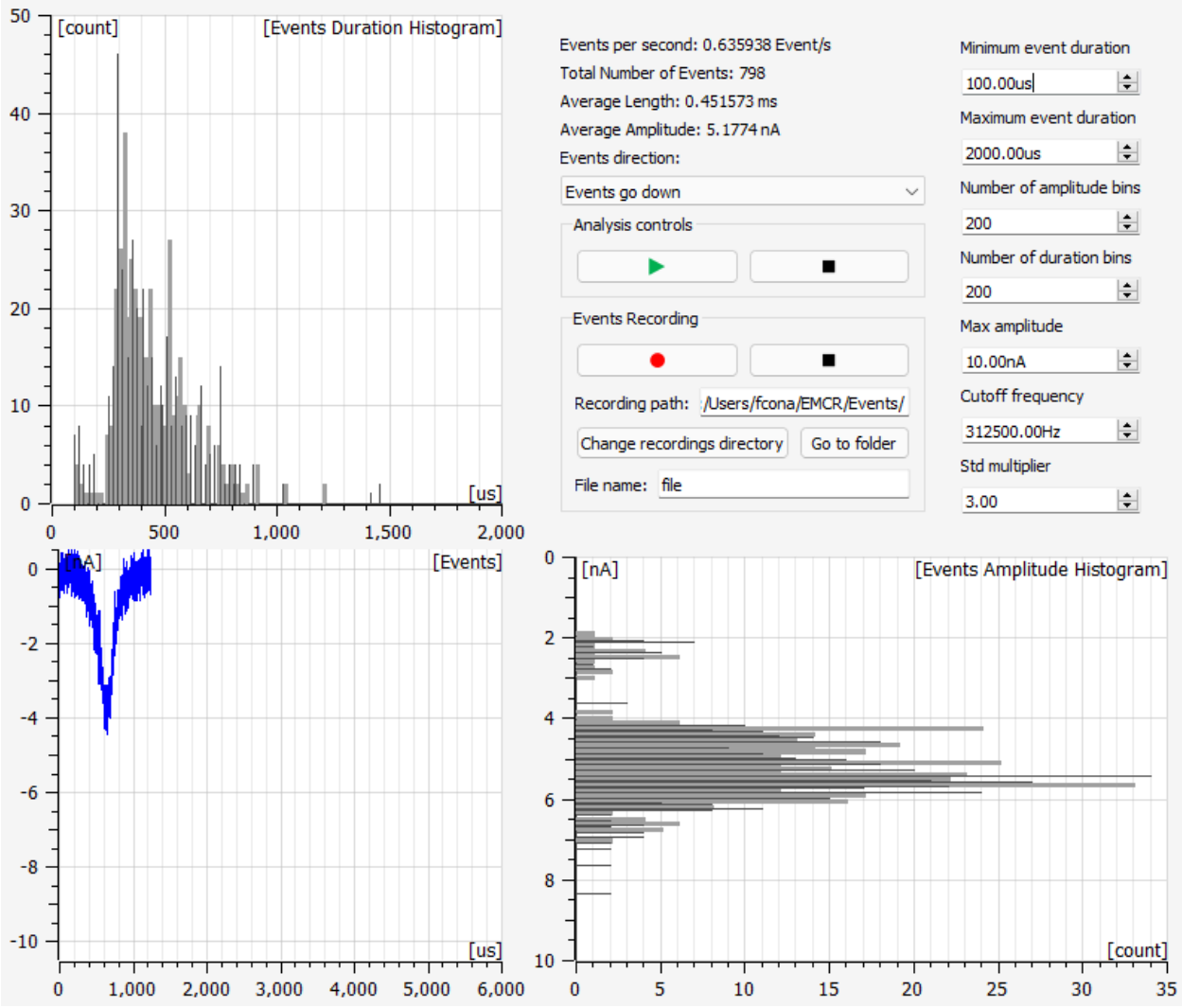
The IV Graph analysis monitors the voltage value and for each value computes the average current. Currently the x axis is divided into **3200 voltage bins**, so for example if a device can

apply voltages in the range ±1600mV the resolution on the x axis will be 1mV, it won't be possible to appreciate changes smaller than this value.

Every time a voltage value is applied, the software calculates which bin better approximates that voltage and the corresponding current is accumulated. Then the average is calculated to draw a [V, I] pair dot in the plot. If a previously observed voltage value is acquired again, the dot corresponding to the same voltage bin is updated with the new current values, taking the average of the new and old current values.

Event Detection

Event detection is a feature that enables users to extract pulses that diverge from the noise or baseline of the signal they are acquiring.



Its main goal is to help scientists to extract only significant information from the signal, discarding the baseline, reducing the overall data size and providing them with meaningful and aggregated information.

The result of the Event detection is an HDF5 file whose structure can be found in Appendix A of this document.

The HDF5 file format is free and open source, and the files can be opened using the free software **HDFView**.

Event Detection Controls

- Start the event detection analysis: a reinitialization of the data will be performed to discard old data from previous protocols
- Stop the analysis: stop the data flow from the device, the plot will not be updated anymore (until the start analysis button is pressed again)
- Start, stop and configure recordings (the topic of recordings is better described in the following paragraph)
- Set the analysis parameters: see the event detection algorithm paragraph
- Have a quick overview of the events statistics

Recording

By pressing the **START** button, a recording of the selected channels will begin.

You can stop the recording manually by pressing the **STOP** button.

The **Recording path** is the base directory in which all the recordings will be stored, you can change it by clicking the **Change recordings directory** or navigate to it by pressing the **Go to folder** button.

To edit the file name, simply change the content next to the file name label.

Scripts

In the EMCR installation folder (default path: C:\Program Files (x86)\Elements - EMCR) there's a script/events hdf5 folder which contains some simple Matlab and Python scripts that show how to load and plot the events from an .hdf5 file.

Event Detection Algorithm

Objectives

The algorithm is designed to:

- Be memory efficient
- Be computationally efficient
- Use less space than a raw recording
- Extract a baseline for the signal

The algorithm is **not** designed to:

- Be 100% accurate
- Yield accurate results with a fast changing stimulus

Explanation

To extract the events the raw signal is filtered twice.

The first time the signal is filtered with a Butterworth first order IIR filter with a 500Hz cutoff frequency.

The result of this filter will be used as the baseline.

The baseline is then subtracted from the current values in order to have zero-centered current values (this is useful because it allows the user to appreciate the real amplitudes of the events without any offsets introduced by the applied voltages).

The resulting signal is then filtered again with the user input cutoff frequency to reduce the high frequency noise. Such cutoff frequency heavily depends on the SR of the experiment. If the user does not know what frequency should be set, a good rule of thumb is to choose **SR/4**, so that the reduction in noise is low but the risk of neglecting real events is minimized.

Each current value is then compared to a threshold value, calculated as **STD*N** where N is the user input parameter Std multiplier, and STD is the estimated standard deviation of the current signal. Greater values for N will lead to less more evident events, while lower values will identify more events but could also save noise instead of real events.

A candidate event is effectively confirmed (and saved to file when the recording is activated) if its duration lies between the input parameters Minimum events duration and Maximum events duration, and if its amplitude is less than the input parameter Max amplitude.

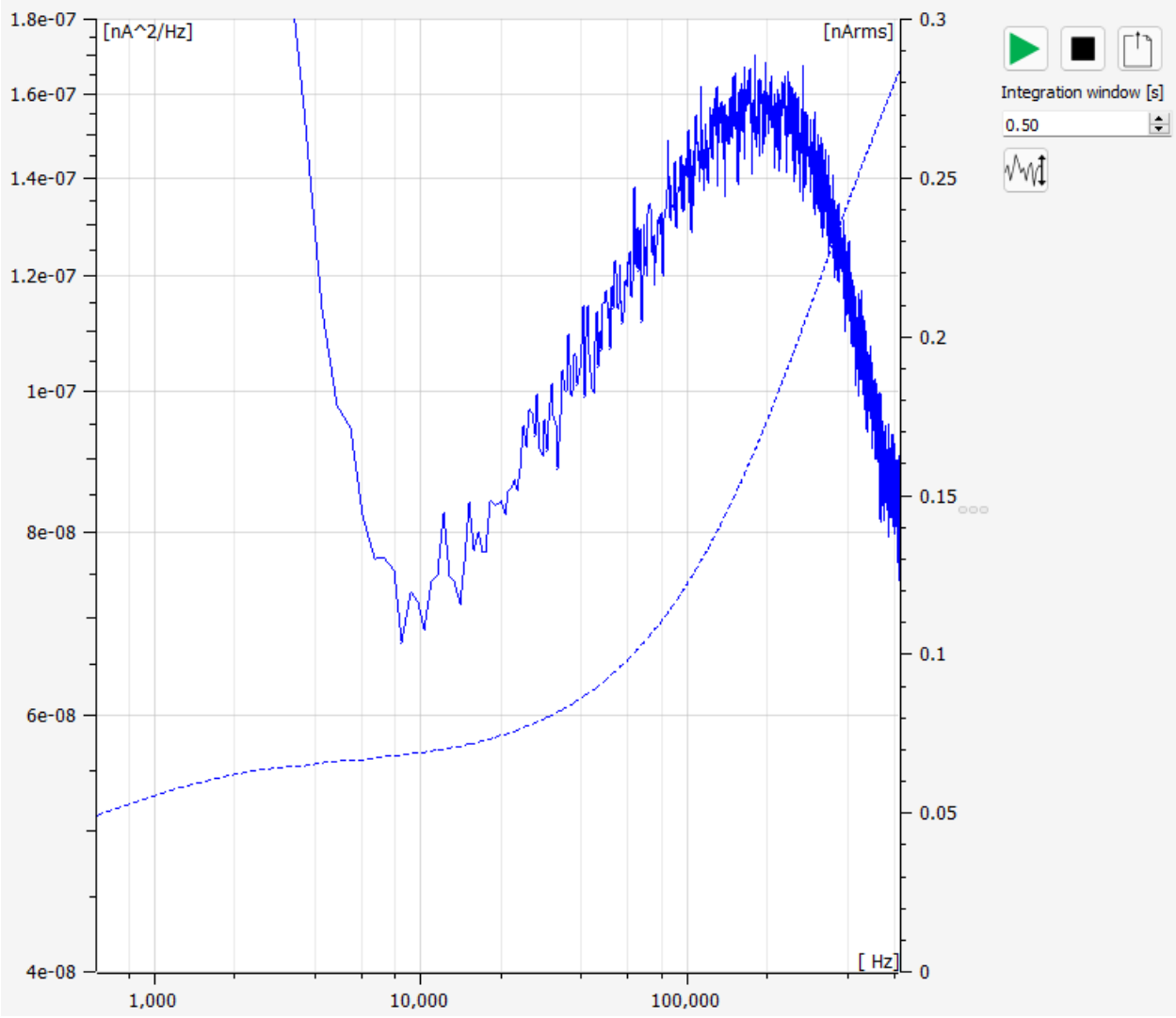
Confirmed events are used to update the events statistics and histograms. To compute the Events Duration Histogram, the time range [0, Maximum event duration] is divided into Number of duration bins (input parameter) subsets. For each duration subset the number of events with the duration such subset are counted and the number is represented with a vertical column. Analogously, to compute the Events Amplitude Histogram, the range [0, Max amplitude] is divided into Number of amplitude bins (input parameter) subsets, and a horizontal line is drawn to represent the events count for each amplitude subset.

There is virtually no limit in the file size generated from this analysis and no overhead when processing them. However, if you need to modify them, make sure to save a copy before any change, otherwise all the data will be overwritten.

Spectrum

The spectrum of a signal and the integral rms (Irms) noise can be estimated and visualized in real time by clicking on the **Spectrum** tab. The spectrum is plotted with a solid line, while the IRms is plotted with a dashed line.

It is important to note that this kind of analysis **only works on channels that are Expanded**.



Spectrum Controls

The right part of this widget will allow the user to perform the following operations:

- Start the spectrum analysis: a reinitialization of the data will be performed to discard old data from previous protocols.

- Stop the analysis: stop the data flow from the device, the plot will not be updated anymore (until the start analysis button is pressed again)
- Export the data: export the plotted points to a CSV file of your choice
- Parameters: see the spectrum algorithm paragraph
- Auto zoom the y axis

Spectrum Algorithm

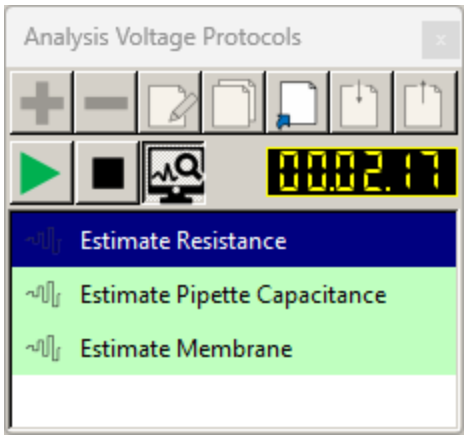
The spectrum is estimated using the periodogram method. An amount of data corresponding to at least the defined integration window is collected. The data is then divided into blocks of 2048 samples each. An FFT is computed for each block and the spectrum is obtained as the average of the squared amplitudes of all the FFTs[1].

The Irms is derived from the spectrum by taking the cumulative sum of the spectrum[2] and then the square root of each component. The last value of the Irms should match the rms value computed by the Measurement overview widget, with a small tolerance due to the fact that the 2 analyses don't necessarily use the same chunks of data.

[1] A scaling factor has to be applied for the spectrum to have physical sense. The scaling factor equals 2/N where N is the total number of samples used for a single estimation, i.e. the number of samples corresponding to the integration window.
[2] The DC component of the spectrum (null frequency) is not taken into consideration for the computation of the Irms.

Protocol Based Analyses

These analyses can be accessed from the protocol widget by clicking the  button.



These analyses are characterized by the fact that their results are scalar and are displayed in the **Measurement Overview widget**, and that they require a very specific stimulation protocol to work, so they start when the corresponding protocol is started.

Measurements Overview

Channel index	Mean Voltage [mV]	Voltage RMS [mV]	Mean Current [nA]	Current RMS [nA]	Resistance [MOhm]	Pipette capacitance [pF]	Membrane capa
1	2.96971	199.978	-0.00363311	0.354104	70988.9		
2	2.96971	199.978	0.00986362	0.262894	87202.3		
3	2.96971	199.978	0.0174801	0.249971	92211		
4	2.96971	199.978	-0.011258	0.389571	90030.8		

The currently available protocol based analyses are:

- Resistance estimation
- Pipette capacitance estimation
- Membrane estimation

Resistance Estimation

Provides an estimation of the resistance, i.e. the impedance at frequency = 0Hz. It requires a square wave voltage protocol.

It is important to note that this kind of analysis **only works on channels that are Expanded**.

Resistance Estimation Parameters

- Vamp: amplitude of the square wave. Choose a value that provides a large current excursion, but avoiding current saturation.
- tpulse: duration of the square wave pulses. Choose a value at least twice as large as the current transients, to ensure that current regime values are computed properly.

Resistance Estimation Algorithm

The resistance is estimated as the delta voltage dV divided by the delta current dI. dV is the peak-to-peak amplitude of the voltage stimulus. dI is computed as the difference between the current regime value during the upper pulse and the current regime value during the lower pulse. The current regime value is computed as the average of the current value from 50% to 75% of the voltage pulse. The estimation is repeated 5 times, the final result is the average of these 5 estimations.

Pipette Capacitance Estimation

Provides an estimation of the pipette capacitance in patch clamp experiments. It assumes that the DUT is a pure capacitance. It requires a square wave voltage protocol.

It is important to note that this kind of analysis **only works on channels that are Expanded**.

Pipette Capacitance Estimation Parameters

- Vamp: amplitude of the square wave. Choose a value that provides a significant capacitive peak, but smaller than the selected current range.
- tpulse: duration of the square wave pulses. Choose a value at least twice as large as the current transients, to ensure that current regime values are computed properly.

Pipette Capacitance Estimation Algorithm

The pipette capacitance is estimated as the electrical charge Q absorbed by the capacitance divided by the delta voltage dV. dV is the peak-to-peak amplitude of the voltage stimulus. Q is computed as the current transient integral in time. The current transient is computed with respect to the current regime value. The current regime value is computed as the average of the current value from 50% to 75% of the voltage pulse. The current transient is acquired 5 times and averaged before being used to estimate the charge.

Membrane Estimation

Provides an estimation of the membrane capacitance, of the access resistance and of the membrane resistance in patch clamp experiments. It assumes that the DUT is a whole cell

model, where the pipette capacitance has already been compensated for. It requires a square wave voltage protocol.

It is important to note that this kind of analysis **only works on channels that are Expanded**.

Membrane Estimation Parameters

- Vamp: amplitude of the square wave. Choose a value that provides a significant capacitive peak, but smaller than the selected current range.
- tpulse: duration of the square wave pulses. Choose a value at least twice as large as the current transients, to ensure that current regime values are computed properly.

Membrane Estimation Algorithm

First of all the total resistance Rt (sum of access resistance Ra and membrane resistance Rm) is estimated using the resistance estimation algorithm. Then the time constant τ (product of Ra and the membrane capacitance Cm) is estimated by fitting a first order exponential to the capacitive peak transient. The membrane capacitance is then estimated as the electrical charge Q absorbed by the capacitance divided by the delta voltage dV. dV is the peak-to-peak amplitude of the voltage stimulus. Q is computed as the current transient integral in time plus the correction charge Qc. The current transient is computed with respect to the current regime value. The current regime value is computed as the average of the current value from 50% to 75% of the voltage pulse. The current transient is acquired 5 times and averaged before being used to estimate the charge. Qc is the integral of the current which is flowing on Rm with respect to the current regime value. This current is estimated as a first order exponential with time constant τ. Ra is obtained as τ/Cm. Rm is Rt - Ra.