**Data Acquisition using *patchclamp***

**1. History**

The Matlab program *patchclamp* (version 1.0, released March 2021) can be used for acquiring patch clamp and field potential recordings from the laboratory’s slice physiology rigs. The data are stored in a format that can be exported to ASCII (allowing for analysis by third-party software such as Origin or pClamp) or that can be analyzed natively using the Matlab program *analyzepatch* (described in a separate document on this site) and Matlab itself.

*patchclamp* grew out an earlier program, called *ni*, that Niraj Desai wrote at The Neurosciences Institute (R.I.P., <https://en.wikipedia.org/wiki/The_Neurosciences_Institute>) and which was described in the supplementary material of Desai et al., J. Neurophysiol. 114:1331-1345 (2015) and at the website [www.robotpatch.com](http://www.robotpatch.com). That early program itself was the result of an even earlier collaboration between Eugene Izhikevich and Niraj Desai at The Neurosciences Institute (years before Eugene left academic science to make his fortune: [www.braincorp.com](http://www.braincorp.com)). The newest incarnation, *patchclamp*, utilizes the latest innovations (as of early 2021) that The Mathworks has incorporated into Matlab’s application building and data acquisition features.

**2. Getting started**

**Requirements**

Software: (1) Matlab R2020b or later. (2) Matlab’s Data Acquisition Toolbox, version 4.2 or later.

Hardware: A data acquisition device compatible with Matlab’s Data Acquisition Toolbox: <https://www.mathworks.com/hardware-support/data-acquistion-software.html>. Most National Instruments multifunction data acquisition devices (such as the PCIe-6343, which is what our rigs contain) satisfy this requirement.

**Installation**

The program *patchclamp* consists of a set of M-files written in Matlab (plus a small amount of supplementary code written in C++). As such, the program works *within* the Matlab environment and does not require a formal installation.

The latest code will always be available on the GitHub site: <https://github.com/nsdesai/patchclamp>. (A nearly up-to-date version will also be on the TEAMS site of the LCSMS.)

Simply download the patchclamp folder to your hard disk. You can put the folder anywhere, but the most sensible place is within the Matlab folder, which you will find in the Documents folder (assuming you’ve already installed Matlab).

Then, let Matlab know where to find these files by adding the folder to its path. For example, type these two lines at the Matlab command line:

>> addpath('C:\Users\niraj\Documents\MATLAB\patchclamp')

>> savepath

Of course, you should substitute the name of your own file path (unless your name happens to be “niraj”).

If any of this confuses you, go here: <https://www.mathworks.com/help/matlab/search-path.html?s_tid=CRUX_lftnav>. Read the topics “What is the MATLAB Search Path?” and “Change Folders on the Search Path.”

**First use**

The very first time *patchclamp* is run on a new computer, we need to make sure that it can interface correctly with the patch clamp amplifiers and the National Instruments (or other) data acquisition device.

In this laboratory (LCSMS), we use Multiclamp 700B amplifiers. These can be controlled by the *patchclamp* program itself. Other amplifiers require entering gain settings manually (and, of course, knowing what they are).

Type this at the Matlab command line:

>> patchclamp

This will open up the main *patchclamp* GUI, which looks like this:

Graphical user interface, application

Description automatically generated

Most likely, a second GUI called *hardware* will also open up. If not, press the button **hardware**at the upper left of the *patchclamp* GUI. The *hardware* GUI looks like this:

A screen shot of a computer

Description automatically generated with low confidence

The top panel (labeled **daq board**) controls the amplifier channel assignments of the data acquisition board.

First, choose the name of the data acquisition board to use. In the picture, a USB-6211 is the installed board, but on the LCSMS rigs the board is a PCIe-6353.

Second, make the channel assignments. Notice that channels are divided into two groups: channels 1-4 may be used for patch clamp recordings, whereas channels 5-8 are limited to field potential recordings and other types of recordings that do not require that the input have an associated output. (In current clamp, the input is the membrane potential which is usually associated with a current injection output. In voltage clamp, the input is the current which is associated with a holding potential output.) Channels 1-4 have analog inputs and analog outputs. Channels 5-8 also have analog inputs but only digital outputs (0 or 5 V).

The National Instruments naming convention for its analog inputs is ai0, ai1, ai2, ai3, and so on. (N.B., National Instruments starts counting from zero, not one.) Likewise, the first four analog outputs are ao0, ao1, ao2, and ao3. The first four digital outputs are port0/line0, port0/line1, port0/line2, and port0/line3.

The simplest choice is to assign the inputs and outputs to the channels in order. So, channel 1 has gets as its input ai0 and has as its output ao0. In this case, the first amplifier channel should be connected like this: the amplifier output (called “scaled output – primary” by the Multiclamp 700B) should go to board input ai0 using a BNC cable, and the data acquisition board output ao0 should be fed using a BNC cable into the amplifier input (called “command input” by the Multiclamp 700B). Likewise, for channels 2-4. (Here, I assume that the rig has four amplifying channels, as it does when it has two dual-channel Multiclamp 700Bs.) The outputs of any other amplifiers (not used for patch clamp) can be connected to data acquisition board inputs ai4-7, and they or any other pieces of equipment (e.g., an Iso-Flex stimulator) requiring a TTL trigger can be connected to board digital outputs (such as, port0/line0, port0/line1, port0/line2, and port0/line3).

Third, press the button **update channels**.

Now, look at the bottom panel (labeled **patch clamp amplifier**). The panel tells the data acquisition program about the patch clamp amplifier(s).

First, choose the channel number (1-4). (You’ll do this individually for all four amplifier channels – or however many you have available for patch clamp.)

Second, using the toggle button group on the left (labeled **amplifier**), choose the Multiclamp amplifier and channel to associate with this channel number. If you are using an amplifier other than Multiclamp for this channel number, choose “manual” instead. If you are using a Multiclamp amplifier, the gain settings on the right will be filled in automatically. If you are using a different amplifier, you must enter the gain settings yourself (these will be available in your amplifier’s manual or possibly written on the amplifier itself).

Third, go through the first and second steps for all of the amplifier channels you plan to use (up to four).

Fourth, press the button **update amplifier**.

Finally, press the button **finalize** at the bottom of the GUI. This saves your choices to the file *hardwareConfiguration.mat*, which is in the parameters\_and\_gui subfolder of the patchclamp folder. (Alternatively, you could press the button **cancel** to abort this process. Nothing will be changed if you do this.)

The next time you open the *patchclamp* program, it will check *hardwareConfiguration.mat* to find out what the hardware configuration of the rig is. You need do nothing more. If you do not switch amplifiers or swap the BNC cables, you should not need to go through these steps again – initiated by pressing the **hardware** button of the main GUI. But mostly, the hardware configuration is a one-time deal.

**3. Setting the recording parameters**

Now, look at the main GUI. The panel **parameters** at top left lets us set the recording parameters.

Graphical user interface, application

Description automatically generated

*experiment number* is a number that designates a recording or a set of simultaneous recordings. Every time you swap electrodes to patch onto a new neuron or group of neurons, you should press the button **new experiment**. This will increment the experiment number by one and set the trial number to one. *trial number* is the number designating each trial for an experiment.

For example, say you simultaneously patch onto two neurons and put both in current clamp. Before doing so, you pressed **new experiment** to indicate that a new experiment (“recording”) was about to start. So, we start at experiment 1 trial 1. Now, you inject a set of current steps into both neurons. The resulting data are saved as experiment 1 trial 1. Then, you change the current steps so that they are bigger or smaller and inject these steps while measuring the neurons’ membrane potentials. These data are saved as experiment 1 trial 2. Now, you put the second neuron into voltage clamp and measure its current while causing the first neuron to fire by injecting current. These data are saved as experiment 1 trial 3. Finally, you are done with these neurons. You swap electrodes and press **new experiment** to advance *experiment number* to 2 and set *trial number* to 1. Now, you can patch onto a new set of neurons. The important point: “experiment” designates a particular set of recordings whereas “trial” designates the different trials for that set of recordings.

The pulldown menu **sample rate** allows you to select the sample rate used to digitize the data. For typical patch clamp experiments, a value of 10000 Hz or 20000 Hz will be fine.

The tabbed windows just to the right allow you to set the status and some of the parameters of individual channels. For example, the settings in the picture below mean that channel 1 is in voltage clamp, with a gain of 10, and a holding potential of -70 mV. (N.B. With only one exception – noted below – all voltages in this program will be in mV, all currents in pA, and all times in msec. So, even though the field **holding**doesn’t specify its units, we should assume they are in mV because this is a potential.)

Graphical user interface, application

Description automatically generated

If the channel were in current clamp, the field **holding**would designate how much DC current to inject into the neuron (in pA).

Channels 1-4 have a toggle switch **enable-disable**. When enabled, the channel is on and in use. When disabled, the channel is off and not used. The reason for having this switch is that, even if your rig is configured for four patch clamp recordings, on some days we may only wish to use one or two or three. The unused channel(s) can be disabled with this switch, so they won’t be used, and the *patch clamp* program will not look for them.

Besides “voltage clamp” and “current clamp”, the other options for **status**are “I=0”, “field potential”, and “off”. When the status is “I=0”, the membrane potential is recorded but no current can be injected. When the status is “field potential”, the field potential from the amplifier is recorded. When the status is “off”, nothing is recorded on this channel. When the status is anything except “off”, the output can still be used: it will be in volts and will not be scaled into picoamps (as in current clamp) or millivolts (as in voltage clamp).

The data acquisition program and the patch clamp amplifier must agree on the state of each amplification channel. To scale the inputs and outputs correctly, the data acquisition program must know (1) whether the channel is in voltage clamp, current clamp, or the I=0 state, (2) what the gain is (how much the amplifier is amplifying), and (3) what holding potential (in voltage clamp) or holding current (in current clamp) has been applied. In the case of Multiclamp amplifiers, which are computer controlled, the program and the amplifier can talk to each other. So you have two possibilities for making sure the program and the amplifier are in agreement: (1) choose the desired settings in the *patchclamp* GUI using the tabbed windows just described and then press the button **write multiclamp**(see ; this writes the settings to the amplifier; or (2) choose the desired settings in the Multiclamp Commander window, which is the software provided by Molecular Devices to control the Multiclamp amplifier, and then press the button **read multiclamp**; this directs the data acquisition program to query the amplifier and read its settings. In the case of non-Multiclamp amplifiers, you must make sure manually that the settings of the *patchclamp* GUI are in fact the same as the settings of the amplifier, which for most non-Multiclamp amplifiers are set by physically adjusting knobs and switches.

Once you have set the parameters for all the channels you wish to use, you can save them for future use by pressing the button **save parameters**, which will open a dialog allowing you to choose a name and folder in which to save the parameters file. In subsequent recording sessions, you can call up these parameters by pressing the button **load parameters** and using the dialog to select your saved parameters file.

Graphical user interface, application

Description automatically generated

**4. Using *test pulse* to get a patch clamp recording**

Most amplifiers have a test pulse feature, which helps you obtain patch clamp recordings. Typically, in voltage clamp, a small voltage step (-10 mV) is applied while you monitor the resulting current deflection on an oscilloscope. You can, of course, still use that method with the *patchclamp* program, but there is another possibility: use the built-in test pulse, which you can find in the panel at top center of the GUI.

Graphical user interface, application

Description automatically generated

The amplifier must be in voltage clamp for the test pulse to work properly. Use the checkboxes to select which channel(s) to monitor. Once the electrode(s) are in the bath, press the button **zero** to zero them (i.e., thus setting the bath potential as the reference potential). Then press the button **test**. A pulse of amplitude (in mV) specified by the field **amplitude** and duration (in msec) specified by the field **duration**will be applied continuously. The resulting current deflections will be plotted in the windows labeled **graph 1**and **graph 2** immediately below the **test pulse** panel.

If you switch to the **resistance & holding**tab, the program will give you a running update on the tip resistance(s) (in MΩ).

Graphical user interface, application

Description automatically generated

After you’ve pressed the electrode against the cell membrane and released the positive pressure, you’ll want to add a holding potential (e.g., -70 mV) to aid in forming a giga-ohm seal and so that, when you break through, the voltage clamp will to a reasonable potential. To do this within the program, press the button **-70 mV** immediately to the right of the desired channel.

**5. Creating outputs**

To create outputs (current injections in current clamp, voltage commands in voltage clamp, TTL triggers, etcetera), use the **outputs** panel at bottom right of the main *patchclamp* GUI.

Graphical user interface, application

Description automatically generated

Data acquisition using *patchclamp* is episodic (as opposed to continuous). The episodes can be quite long (>10 min), but you must specify the output for the whole episode.

The fields of the **outputs** panel include duration (how long each episode is), period (the time between the start of one episode and the start of the next episode), repetitions (how many times to repeat the set of episodes specified in the panel), and shuffle (if the specified episodes differ from each other – for example, a family of current steps where the amplitude is being varied – shuffle “on” means randomize the order of in which the episodes are presented).

There are 12 lines in which different kinds of stimuli can be specified. What goes to a particular channel depends on which checkboxes are checked at left. In the picture shown above, channel 1 gets the output on line 1, whereas channel 2 gets the outputs on lines 6-9. When a channel is marked for two or more lines, the outputs of those lines are summed together. So, channel 2 gets a cosine output between t=1500 msec and t=2500 msec \*and\* it gets three current steps that begin at t=4000 msec, 4200 msec, and 4400 msec, respectively. Because those current steps overlap in time, they sum up to a staircase. (The outputs are plotted in the **output waveforms** tabbed window above the **outputs** panel. After you make changes to the outputs, press the button **update outputs** to see the results plotted out.) All eight channels (the four patch clamp channels and the four other channels) can be run simultaneously.

Each is specified by three fields: **Time**, **Duration**, and **Amplitude**. Time is the time (in msec) that the stimulus starts. Duration is the how long it lasts (also in msec). Amplitude is the description of the stimulus itself. Amplitude can be a family of current steps (specified as a Matlab row vector) as on the first line of the example above, a single number as in the third line, a Matlab function as in the fourth and sixth lines (a ramp and a sinusoidal function, respectively), a function specified by the user in an M file (as in the eleventh line, which references a custom function that produces Ornstein-Uhlenbeck noise), or an arbitrary set of numbers saved to an MAT file on the Matlab path or a variable in the current workspace.

In the subfolder examples within the patchclamp folder, you will find a wide variety of examples of output stimuli. These give an idea of the possibilities.

Once you have created the outputs, you can save them for future use by pressing the button **save outputs**. This will open a dialog box that will allow you to specify the file name and folder in which to save the output parameters. Subsequently, you can load a previously-created set of outputs by pressing the button **load outputs** and navigating to your saved file.

**6. Running experiments**

You can start and stop recordings using the controls of the **recording** panel at left of the main *patchclamp* GUI.

Graphical user interface, application

Description automatically generated

Pressing the **start** button will begin episodic acquisition. The results will be plotted on the axes **graph 1** and **graph 2** just to the right of the **recording** panel. If you wish to stop the recording before it completes, press the same button again.

If the **save data** checkbox is clicked, data will be saved within the Matlab folder in the Documents folder. In particular, when it is started, the program creates a data subfolder. If the date is March 14, 2021, then the subfolder will be C:Users\niraj\Documents\MATLAB\2021\Mar\14. For experiment 3, trial 4, the data file will have a base name of experiment003trial004.mat. It will contain the measured responses in a variable inputData, the stimuli in a variable outputData, and the recording parameters (e.g., sample rate, channel gain, absolute time) in a variable Pars.

If the **save data** checkbox is not clicked, the data will still be saved but in a folder within the data folder called “temp”. The file name in that case will also include a time stamp. The rationale for saving data anyway, even if the user has unclicked the checkbox, is in case they did so by mistake or they change their mind later – the data can still be recovered.

The field **descriptor** is appended to the base name of the saved file. So, if the field reads “\_before\_drug\_application”, then the data of experiment 3, trial 4, will be saved as experiment003trial004\_before\_drug\_application.mat – to allow users, while doing analysis, to quickly determine what the file is.

Any notes written in the field **notes** will be appended to the variable Pars saved in the data file and can be viewed during the analysis. The field therefore is a place to write out any details you want saved with the data.

Also saved with the data, in the variable Pars, will be the **mouse ID** and its date of birth (**DOB**), if you fill in those fields, along with its sex **female-male**.

**7. Seeing the results**

As the data are acquired, they are plotted on the axes **graph 1** or **graph 2** of the main *patchclamp* GUI.

Graphical user interface, application

Description automatically generated

You can choose which graph each of the eight possible channels uses by checking the boxes at the bottom.

Some people find the **graph 1** and **graph 2** axes to be too small. Those people can toggle the **popup** switch to on. If they do, the plots will appear in a new larger popup window.

Chart

Description automatically generated

The x and y scales can be adjusted using the sliders on the bottom and on the left, respectively.

**8. Real time analysis**

Real time analysis – meaning, analysis during the recording session – is possible for some things. The subpanel **analysis**is at lower left of the main *patchclamp* GUI.

Graphical user interface, application

Description automatically generated

If you select *check* in the **stability?** button, then the program will constantly measure and plot a neuron’s input resistance and resting potential (or holding current) and a recording’s series resistance. This will appear at upper right of the main *patchclamp* GUI in the tabbed window **stability**.

Graphical user interface, application

Description automatically generated

Other types of real time analysis (e.g., EPSC amplitude) are available by selected them from the **other analyses** pulldown menu in the **analysis** subpanel.