patchclamp documentation

Version 02/20/22

Niraj S. Desai

niraj.s.desai@gmail.com

This is a first draft of documentation for *patchclamp*, a data acquisition program for cellular neurophysiology (see https://github.com/nsdesai/patchclamp). The program itself is still under revision and this document will certainly be revised, but it gives a good idea of how the program is structured -- including its fundamental logic -- and how to use it.

Please direct any questions or comments to me (NSD) at niraj.s.desai@gmail.com.

Table of Contents

Requirements

- **Windows.** A Windows computer is required because the software makes use of Matlab's Data Acquisition Toolbox, which is available for Windows operating systems only. *patchclamp* has been successfully tested on both Windows 10 and Windows 11 systems.
- **Patch clamp amplifier.** The program was written with the Multiclamp 700A and 700B patch clamp amplifiers in mind, but it can, in principle, be used with any amplifier. However, computer control is available only for Multiclamp amplifiers. Soon I will add additional support to read the telegraph outputs of the Axopatch 200B, but full control is limited to the Multiclamp models because (a) many existing amplifiers, including the Axopatch 200B, cannot be computer controlled and (b) I don't have access to the other amplifiers that can.
- **Data acquisition device.** The program was written with National Instruments devices in mind, but any device supported by Matlab's Data Acquisition Toolbox should work. A list of supported devices is available here: https://www.mathworks.com/hardware-support/data-acquistion-software.html.
- **MATLAB.** The program requires Matlab and its Data Acquisition Toolbox. Any Matlab version more recent than Matlab 2018a should work, but I recommend Matlab 2021a or later. The Mathworks significantly improved the graphics capabilities of its App Designer feature with that release.

Installation

Software

- Install your patch clamp amplifier and data acquisition device according to the manufacturers' instructions. For example, (1) attach a Multiclamp 700B amplifier to your computer using a USB cable and then download and install the Multiclamp Commander software from the Molecular Devices website: https://support.moleculardevices.com/s/article/Axon-MultiClamp-700B-Commander-Download-page and (2) place a National Instruments PCIe card in your computer and download and install the NI-DAQmx software from the National Instruments website: https://www.ni.com/en-us/support/downloads/drivers/download.ni-daqmx.html#428058.
- Download and install Visual Studio Community 2019 from Microsoft: https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes. During the installation, select all the options related to C++ programming. (If you are unsure of what to select, just write to me.) This step is necessary because, to control Multiclamp amplifiers, I wrote a separate program as a Visual Studio C++ project; the Matlab

program *patchclamp* communicates with the amplifiers through this C++ program. Do not install Visual Community 2022 instead of 2019. While 2022 is the newest version of Microsoft's community offering, at this writing (February 2022) most community software, including Matlab, are not ready to use it.

- Download and install Matlab and its Data Acquisition Toolbox. (If you work at NINDS, you can do so through NINDS IT, as we have a site license.)
- Within Matlab, open the Add-On Explorer (look for "Add-Ons" on the toolbar at the top, look on the right). Find and install the Data Acquisition Toolbox Support Package for National Instruments NI-DAQmx Devices.
- Download the *patchclamp* folder from https://github.com/nsdesai/patchclamp and put it in the Matlab folder within the Documents folder.
- Add the folder to the Matlab search path. You can do this in multiple ways. One way: with the Matlab folder as the current directory, type the following lines at the Matlab command line:

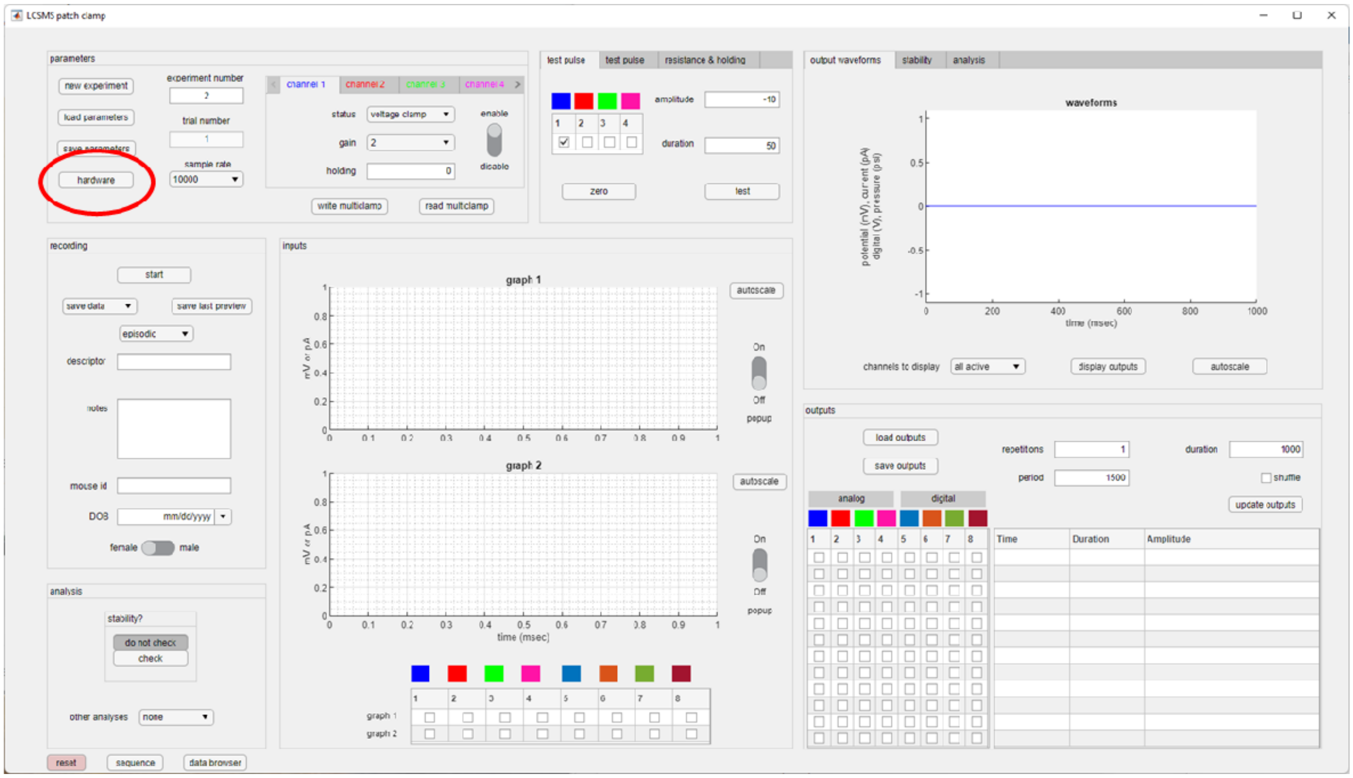
```
addpath('patchclamp')
savepath
```

- Finally, connect the amplifier and the data acquisition (DAQ) device with BNC cables. How you do this is up to you but a sensible arrangement is to connect the first output channels of the amplifier (called "Scaled Output" in the case of Multiclamp amplifiers) to the first input channels of the DAQ device (called "ai0", "ai1", and so on in the case of National Instruments devices). Likewise, connect the first input channels of the amplifier (called "Command Input" in the case of the Multiclamp amplifiers) to the first output channels of the DAQ device (called "ao0", "ao1", and so on in the case of National Instruments devices). If you're planning to use the National Instruments digital outputs to trigger electrical or optogenetic stimulation, connect those too. For example, connect the first digital outputs (called "port0/line0","port0/line1", and the like in the case of National Instruments devices) to the target stimulation devices, such as an IsoFlex stimulus isolation unit or an LED optostimulation unit.

Specifying connections

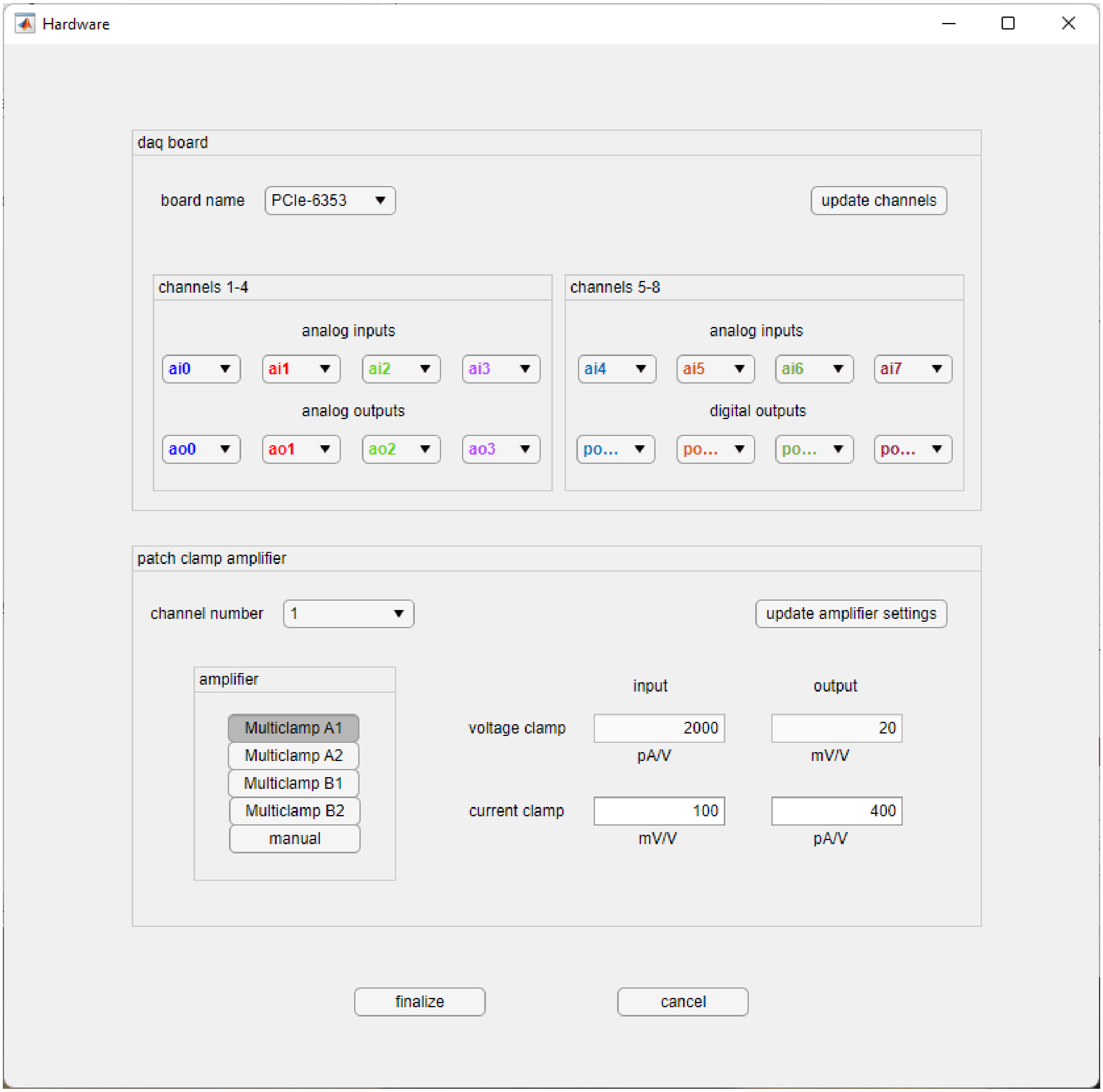
The first time you open *patchclamp* is special in that you need to start by telling the program what is connected to it. That is to say: what is the patch clamp amplifier? what is the data acquisition device? what channels are connected to what?

To answer these questions, start by first turning on any amplifiers you have, and then opening the *patchclamp* program by typing "patchclamp" at the Matlab command line; this will open up the main *patchclamp* graphical user interface (GUI). (It is important that you turn on the amplifiers before opening the *patchclamp* program because the program searches for connected amplifers at startup.)



At upper left (circled in red) is a button called "hardware". Push it.

Doing so opens up a different GUI, one also called "hardware".



The top panel specifies the connections of the DAQ board. (In this example, it is a PCIe-6353 from National Instruments). The bottom panel specifies the connections of the patch clamp amplifier. (In this case, two Multiclamp amplifiers are installed). The entries will come pre-populated with the most likely connections, but one can make selections for both DAQ board and patch clamp amplifier through this GUI.

Top panel. The program is set up to handle up to eight (8) channels. The first four (channels 1-4) are meant for patch clamp connections. Each connection includes both an input and an output. In the case of a Multiclamp 700B, the input specifies which DAQ analog input receives a signal from the "Scaled Output" of the Multiclamp, and the output specifies which DAQ analog output sends a signal to the "Command Input" of the Multiclamp. In the example above, a Multiclamp 700B is using the National Instruments board's first analog input (ai0) to read its scaled output and its first analog output (ao0) to send it commands (e.g., a current injection or a holding potential).

The second four (channels 5-8) are meant for digital connections (i.e., triggers for stimulus isolation unit, LED light, picospritzer). These triggers are sent by the National Instrument board's digital outputs, which have names like port0/line0, port0/line1, and so on. These channels can also be used to record analog inputs from sources other than a patch clamp amplifier, such as an extracellular amplifier for field potentials or a pressure monitor. In this example, National Instruments analog inputs ai4, ai5, ai6, and ai7 are used in this way.

After making changes to these assignments, press the button *update channels* to cofirm your changes.

Bottom panel. Each of the four patch clamp channels can be selected through the dropdown menu *channel number*. The toggle buttons allow you to assign a given patch clamp channel to one channel of a Multiclamp amplifier (each of these amplifiers has two channels) or to specify a "manual" amplifier (i.e., one not controlled by *patchclamp*).

In the case of Multiclamp amplifiers, the program will query the amplifiers and determine the correct scaling for current clamp and voltage clamp recordings. For example, when the amplifier is in current clamp, it signals the membrane potential (*Vm*) through a voltage signal sent out of its Scaled Output BNC. The signal will be a voltage equal to *gain* * *Vm* / (100 mV/V), where *gain* is the gain a user specifies in the Multiclamp Commander window. If you specifed a *gain* of 10 in the Commander and the neuron is sitting at *Vm* = -65 mV, then the signal coming out of Scaled Output will be 10 * -65 / 100 = -6.5 V.

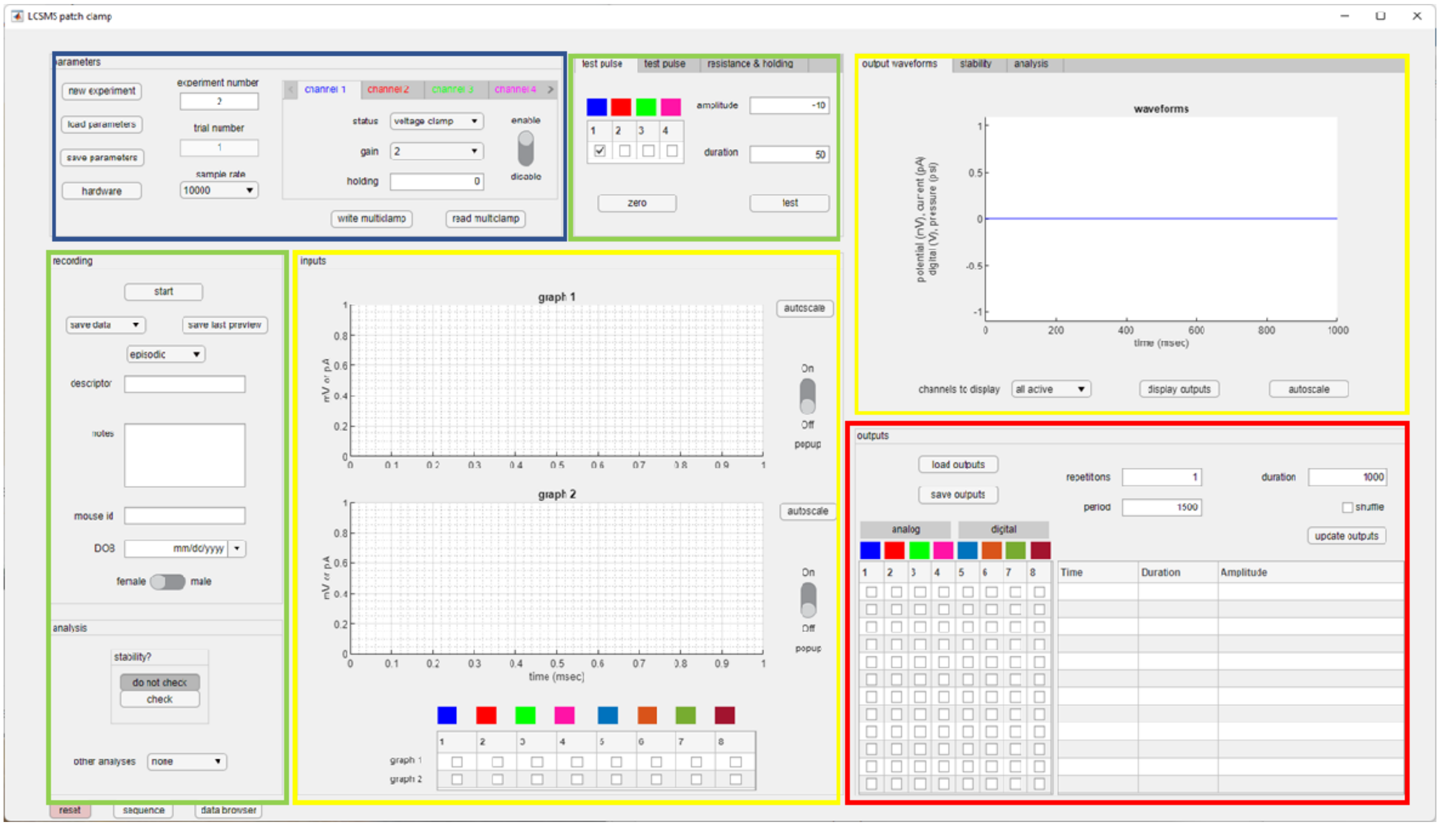
Different amplifiers have different scalings. The Multiclamp 700A's current clamp setting, for example, differs from that of the Multiclamp 700B by a factor of 10. If you use an amplifier other than a Multiclamp (or, as soon as I write the revisions, an Axopatch 200B), then you need to know your amplifier's scalings and enter them into this GUI. (In some cases, the scalings will be written on the amplifer itself, next to the corresponding BNC port.)

After making changes to these assignments, press the button *update amplifier settings* to confirm your changes.

Finally, press the button *finalize* at the bottom to save all changes and close the *hardware* GUI. The hardware configuration will be saved in the file hardwardConfiguration.mat, which may be found inside the folder patchclamp > parameters_and_gui. Anytime after this, when you open *patchclamp*, the program will load this configuration and use it. If you don't add an amplifier or change what is connected to what, you shouldn't need to use the *hardware* GUI again.

The Graphical User Interface (GUI)

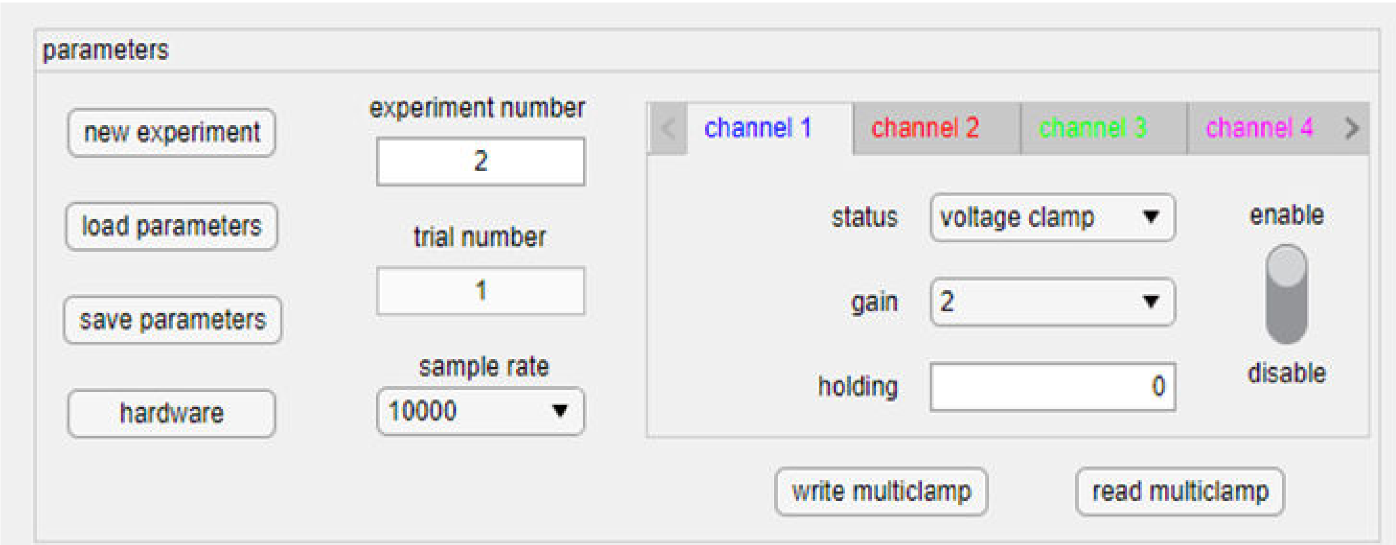
The program works through the main graphical user interface (GUI). It is important to familiarize yourself with its parts. Turn on the patch clamp amplifiers and then open the GUI by typing "patchclamp" at the Matlab command line. (If you reverse the order, the program will complain that it cannot find the amplifiers. In that case, turn on the amplifiers and type "patchclamp" again. It should be fine.)



As illustrated, the GUI is divided into four main areas: (1) parameters, blue box; (2) outputs (a.k.a. stimuli or protocols), red box; (3) controls, green boxes (one is at left, one is at upper middle); and (4) displays, yellow boxes. These are fundamental categories in all types of data acquisition, and it is useful to go through them in some detail to understand the logic of the program and how to navigate the GUI.

Parameters

The section of the GUI at upper left deals with "parameters". In specifying parameters, one specifies things like (a) is the recording being done in current clamp or voltage clamp (or I = 0 mode or field potential mode or something else)?, (b) what is the amplifier gain?, (c) what is the holding potential (voltage clamp) or holding current (current clamp) of the recording?, (d) what are the experiment and trial numbers?, (e) what is the sample rate?



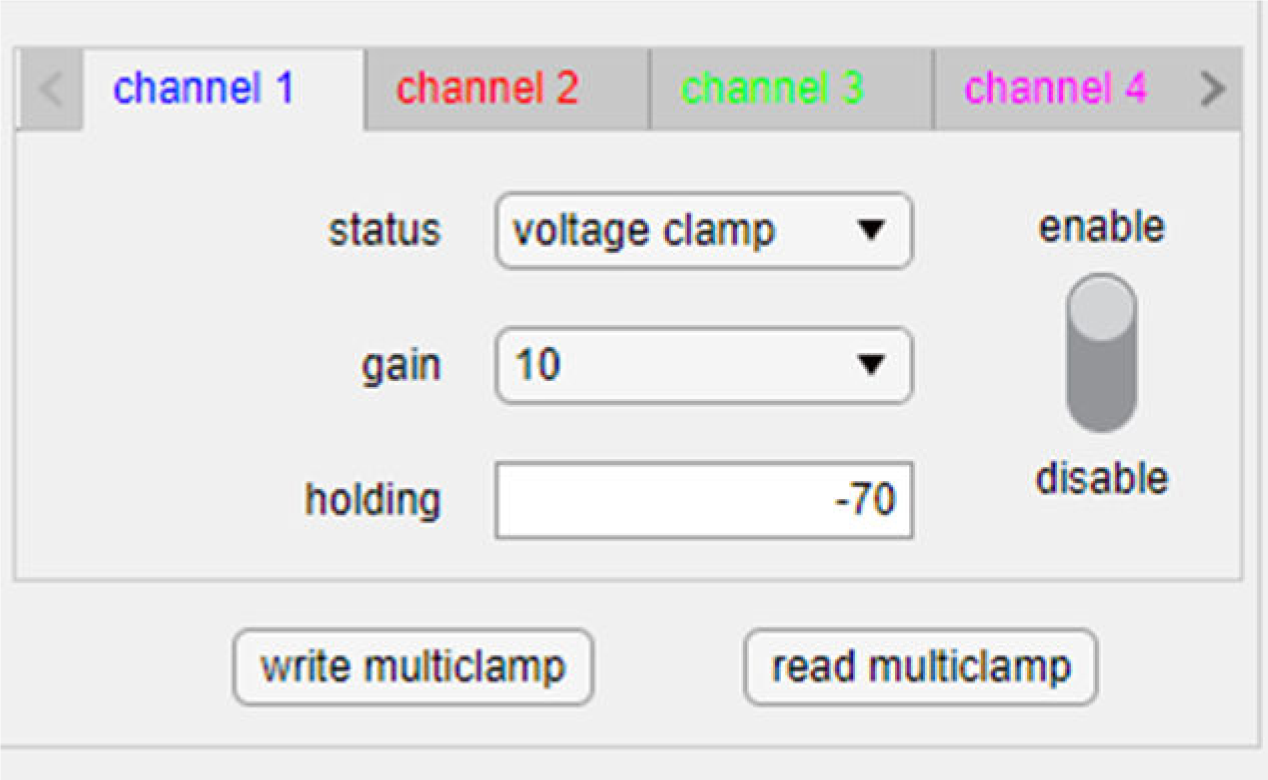
Let us look at the *parameters* subpanel of the GUI.

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 equal to one. *trial number* is the number designating each trial (a.k.a. sweep) 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.)



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** (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

Multiclamp amplifiers, and then press the button **read multiclamp**; this directs the data acquisition program to query the amplifier and read its settings.(Worst case scenario: if you neglect to do either thing, then when you press the **start** button to take a recording, *patchclamp* will spend about 1.5 sec checking and correcting the discrepancy: not a huge penalty but an annoying one.) 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.

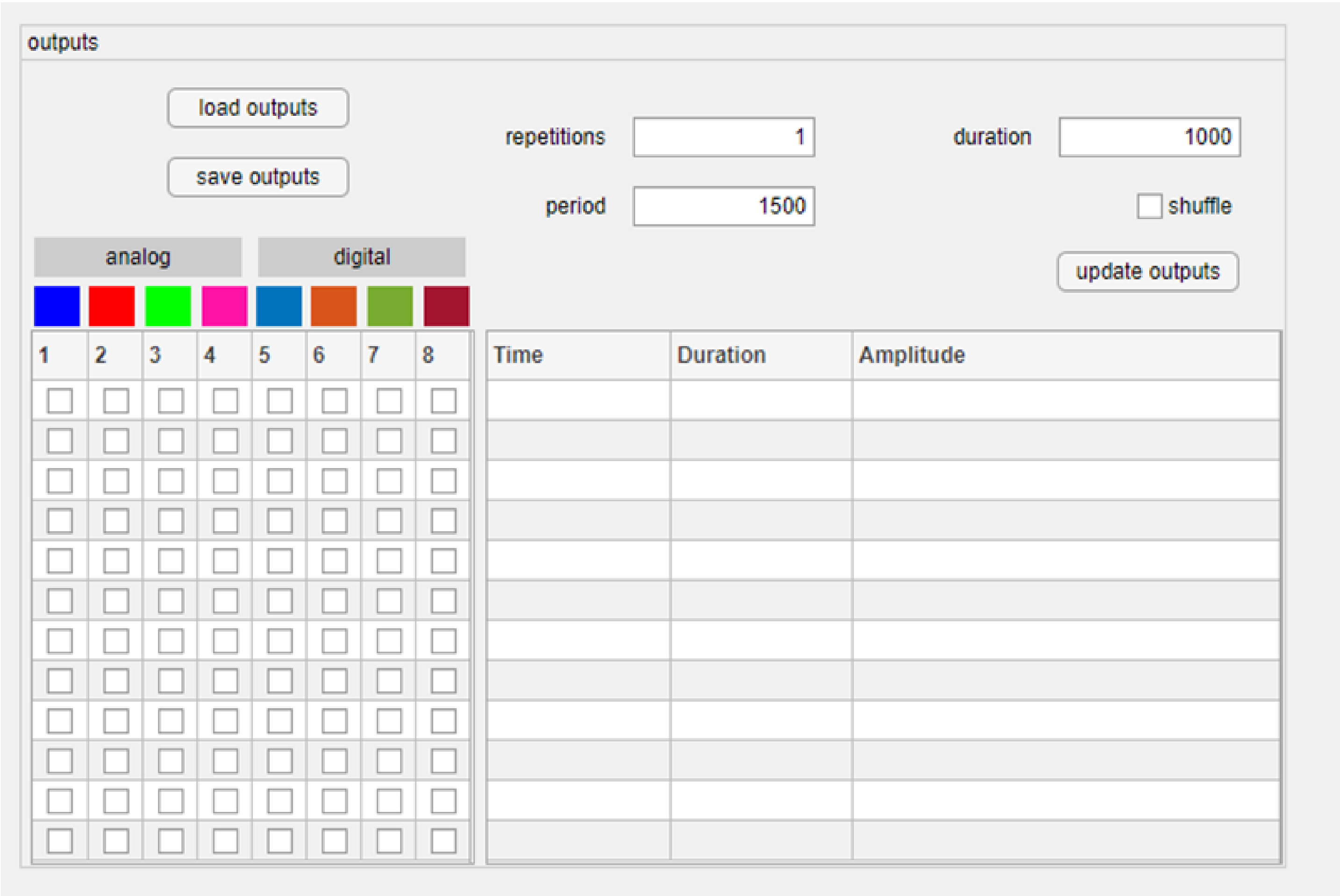
Once you have set the parameters for all the channels you wish to use, you can save them for future use by pressing the button **save parameters**, which will open a dialog allowing you to choose a name and folder in which to save the parameters file. In subsequent recording sessions, you can call up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

Channels 5-8. The other four channels (channels 5-8) simply record voltages straight (without attempting to convert them to mV or pA). They also send out digital outputs (0 or 1). Whereas the outputs of the patch clamp channels (1-4) are tied to the recording mode (current clamp means pA, voltage clamp means mV), the outputs of the digital channels are always 0 or 1 (physically, given the TTL standard, this means 0 V or 5 V).

Outputs

Outputs are different from settings. Settings only specify the state of the amplifier channels (voltage clamp or current clamp or I=0, gain, holding). Outputs are the protocols or stimuli that are injected into the channels (which is to say, neurons or stimulation devices). They include current steps and ramps (in current clamp), command potential steps (in voltage clamp), the patterns of triggers sent to stimulus isolation units or LED light sources, and so forth. In other programs, outputs might be called "protocols" or "stimuli". But they're all the same thing.

In *patchclamp*, one specifies an output protocol using the lower right panel of the GUI.



Most data acquisition in *patchclamp* is **EPISODIC.** (A new continuous or gap-free method of recording has recently been added and is described in a subsequent section.) The basic parameters of episodic acquisition are how long each episode of data will be acquired (specified by the field *duration* in msec), how much time will lapse between the start of each episode (specified by the field *period* in msec), and how many times the specified episodes will repeat (specified by the field *repetitions*). If the checkbox *shuffle* is checked, the order of presentation of different episodes will be randomized.

Eight channels of outputs are possible. The first four (analog) are linked to the four patch clamp channels. This means, for example, that if channel 1 is in current clamp and channel 2 is in voltage clamp, the output of the first will be a current in pA whereas the output of the second will be a command potential in mV. The last four (digital) are purely digital outputs -- they can send out 0 V or 5 V.

The outputs panel has 12 lines, which are summed up to form the final output; this allows the user to specify complicated output patterns. Whether a given line applies to a given channel depends upon whether the channel's checkbox for that line is checked or not.

Displays (outputs)

Controls

Displays (inputs)

Popup windows

Zooming & panning

Data browser

Real-time analysis

Sequences & shortcuts

Acquiring Data

Starting *patchclamp*

Getting a gigaohm seal and breaking through

Fundamentals of acquiring data

Example 1: a family of current steps

Example 2: paired-pulse responses with varying time intervals

Example 3: mEPSC recordings

Example 4: inducing LTP with TBS

Example 5: inducing LTP with STDP

Example 6: probing frequency response with sinusoids & chirps

Example 7: characterizing a sodium current

Example 8: feeding synaptic recordings back in current clamp

Example 9: modeling "in vivo-like" background using Ornstein-Uhlenbeck processes

Example 10: probing synaptic summation with alpha functions

Saving Data

Analyzing and Exporting Data

analyzepatch

Exporting to CSV

Exporting to ABF