G-clamp User's Guide

version 1.2

Paul H.M. Kullmann and John P. Horn

Department of Neurobiology and Center for the Neural Basis of Cognition
University of Pittsburgh School of Medicine
E 1440 Biomedical Science Tower
Pittsburgh, PA 15261
http://www.neurobio.pitt.edu

To download this document and the G-clamp program go to:
(http://hornlab.neurobio.pitt.edu)

Contact Information:

Dr. Paul H.M. Kullmann
Phone: 412-648-9291
pkullman@pitt.edu

Dr. John P. Horn
Phone: 412-648-9429
jph@pitt.edu

Grant Support: This document and the G-clamp software that it describes were both developed at the University of Pittsburgh with support from National Institutes of Health grant RO1 NS21065.

First Edition 2003, Revised, October 1, 2004

Table of Contents

1. INTRODUCTION 4

2. SYSTEM REQUIREMENTS 5

3. INSTALLATION OF G-CLAMP 6

3.1. LabVIEW-installation 6

3.2. PXI-controller: Network and software configuration 7

3.3. Configuration of the multifunction I/O card 7

3.4. Configuring access to the remote system 8

3.5. G-clamp download and installation 8
3.5.1. Embedded controller 9
3.5.2. Host computer 9

3.6. Upgrading G-clamp to run under LabVIEW 7 10

3.7. Running G-clamp for the first time 12

4. USING G-CLAMP 13

4.1. Starting G-clamp 13

4.2. The G-clamp menu 14
4.2.1. File > Print Window… 14
4.2.2. File > Preferences > Data Directory 14
4.2.3. File > Preferences > PXI Target 14
4.2.4. File > Preferences > DAQ Board Configuration 15
4.2.5. File > Exit 18
4.2.6. Function > IV Relation; Gsyn Threshold; Synaptic Gain 18

4.3. Controls and indicators on the G-clamp user interface 19

5. EXPERIMENT MODULES 25

5.1. IV Relation 25

5.2. Gsyn Threshold 27

5.3. Synaptic Gain 31

5.4. V-clamp **34**

6. VIRTUAL CONDUCTANCES **38**

7. TEMPLATE FILES **42**

8. DATA FILES **43**

8.1. Names **43**

8.2. Content **43**
8.2.1. IV Relation - *.IVB 44
8.2.2. Gsyn Threshold - *.GTH 44
8.2.3. Synaptic Gain - *.GAI; *.GAJ 45
8.2.4. V-clamp - *.PKP 46

8.3. Save locations **46**

9. ADDING NEW MODULES TO G-CLAMP **47**

9.1. Adding a new conductance module **47**

9.2. Adding a new experiment module **51**

10. TROUBLE SHOOTING **51**

10.1. Jitter spikes of approximately 250 µs on the PXI controller while running LabVIEW RT **51**

10.2. Poor Network Performance in LabVIEW RT When Connected to Another Computer via a Crossover Cable **52**

10.3. Sometimes saving of very long traces on the PXI controller fails **53**

11. KNOWN BUGS **53**

1. Introduction

G-clamp is a high performance dynamic-clamp system that can implement virtual ionic conductances in living cells. The system was designed for flexibility and ease of use. It can mimic multiple channel types including voltage-dependent sodium and potassium channels and ligand-gated ion channels. Properties of these virtual channels (magnitude, gating, selectivity) can be modified through simple menu selections and program changes. The program can also read conductance template files that reflect complex patterns of synaptic activity. Further details describing the dynamic clamp method, the components of G-clamp and the system's performance can be found in:

> Kullmann, P.H.M., Wheeler, D.W., Beacom, J. and Horn, J.P.(2004) Implementation of a fast 16-bit dynamic clamp using LabVIEW-RT. **Journal of Neurophysiology**, 91: 542-554, 2004.

The G-clamp software application was written in the LabVIEW-RealTime programming environment and requires a Windows host computer and additional hardware. LabVIEW-RT and the additional hardware are products of the National Instruments Co. (www.ni.com). **In order to use G-clamp, you will need to obtain the appropriate software licenses and hardware from National Instruments** (educational pricing is available).

Finally, and perhaps most important, you need not be a programmer to use G-clamp. This manual describes how to install and use the program. This does not require extensive knowledge of LabVIEW. For those users who wish to modify the program, a separate programmer's guide is available at http://hornlab.neurobio.pitt.edu.

2. System Requirements

G-clamp was written in LabVIEW-RT 6.1. It requires at least the Full Development System of LabVIEW plus the LabVIEW-RT 6.1 Real-Time Module. G-clamp will also run under Labview-RT 7 (see section 3.6)

G-clamp uses a few dynamic link libraries (DLLs) and virtual instruments (VIs) from National Instruments that are not part of the LabVIEW-RT 6.1 distribution package[1]. For convenience, these files are included with the G-clamp software that can be downloaded from Horn Lab website (http://hornlab.neurobio.pitt.edu).

The host computer requirements and for running LabVIEW-RT on a PXI controller are described on the National Instruments website (www.ni.com). As a general rule of thumb it is important to note that the speed at which data are displayed on the host computer will depend on the host's processor speed and memory: Generally, the more of each, the better.

[1] These files can be directly downloaded from the National Instruments website and some of them are included in the recently released LabVIEW 7.

3. Installation of G-clamp

This section provides the protocol for a complete first-time installation and configuration of LabVIEW-RT and G-clamp on a PXI-8176 controller with a PXI-6052E multifunction I/O board and a host computer running WindowsXP. If your PXI controller and host computer have LabVIEW-RT already installed and configured, you can jump to section 3.5. G-clamp installation.

3.1. LabVIEW-installation

- Insert LabVIEW 6.0 Full Development System CD into CD drive of host computer.
- Select 'Install LabVIEW' and follow the instructions of the LabVIEW 6i Installation Wizard:
    - Installation Type: Complete
- After the installation has successfully completed, remove CD from drive and restart the computer.
- Insert LabVIEW 6.0.3 Real-Time Full Development System CD into CD drive.
- Install LabVIEW Real-Time 6.0.3 and NI DAQ 6.9.1 (Neither the PID Control Toolset 5.0 nor VISA 2.5.2 and Motion Control 5.1 are required for G-clamp and probably can be omitted. They will be replaced by newer versions later on anyway with the upgrade to LabVIEW 6.1).
- After the installation has successfully completed, remove CD from drive and restart the computer.
- After the restart use the upcoming LabVIEW Real-Time 6.0.3 dialog to mass compile the VIs in the LabVIEW directory.
- Insert the LabVIEW 6.1 Full Development System CD into the CD drive.
- Select 'Install LabVIEW' and follow the instructions of the LabVIEW 6i Installation Wizard:
    - Installation Type: Complete
- After the installation has successfully completed, remove CD from drive and restart the computer.
- Insert LabVIEW 6.1 Real-Time Module CD into CD drive.
- Select 'Install LabVIEW Real-Time' and follow the instructions of the LabVIEW Real-Time 6.1 Installer:
    - Installation Type: Complete
- Connect the host computer and the PXI controller, either via a local area network (LAN) or directly via a CAT-5 crossover cable[2].

[2] Instead of using a CAT-5 crossover cable, a direct peer-to-peer connection can also be established with normal network cables and a hub in between host computer and PXI-controller. This solution avoids the potential problem of incorrect negotiation of the duplex state (half/full duplex) between the controller's Intel ethernet hardware and the host computer's ethernet hardware (see section 11.2. for more information on this problem).

3.2. PXI-controller: Network and software configuration

- On the host computer start the Measurement & Automation Explorer (MAX), which was installed during the previous LabVIEW installation.
- Expand or double-click 'Remote Systems' in the configuration field.
- After MAX has detected the PXI-controller, it shows up with its serial number as its name. Its status will be 'Connected – Unconfigured', all the fields for network settings will be blank, and all the software available on the host computer will not be present on the remote device.
- Network Settings:
    - Because G-clamp uses a fixed IP address for communication with the PXI-controller, changing IP addresses obtained from a DHCP server are not allowed. Instead the IP address has to be set.
    - The subnet mask has to be identical to the subnet mask used by the host computer. The setting on the host computer can be found under Start > Settings > Network Connections > Local Area Connection > Properties > Internet Protocol (TCP/IP) > Properties.
    - Gateway and DNS server can be kept at the defaults 0.0.0.0

  It is not recommended to use the 'Suggest Values…' function, as this function will not obtain the correct subnet mask.
    - Enter a meaningful host name, especially if the connection to the host computer is via a local area network in which more than one PXI-controller might be present.
- Press the 'Apply' button to make the new network settings permanent. MAX will reboot the PXI-controller. (If the PXI-controller is connected to the host computer directly via a crossover cable and its status remains 'Disconnected' after the reboot, see the section 11.2. 'Poor Network Performance in LabVIEW RT When Connected to Another Computer via a Crossover Cable')
- Software:
  If the PXI-controller came with software pre-installed by National Instruments it should now show up as present on the remote device. If not or if the version on the remote device is older than on the host computer, install/upgrade software on the remote device by right-clicking on any software item and selecting 'Install software…'

3.3. Configuration of the multifunction I/O card

This is also done from the Measurement & Automation Explorer:

- Select Tools > NI-DAQ Configuration > Remote DAQ Configuration… and enter the IP address of the PXI-controlller.
- Under 'NI-DAQ Devices' highlight the multifunction I/O card and go to 'Properties':

  - Under 'System' check whether the 'Device Number' displayed is the same as in the previous window.
  - Under 'AI' set 'Polarity/Range' to –10.0V - +10.0V and 'Mode' to Nonreferenced Single Ended
  - Under 'AO' set 'Polarity' to Bipolar
  - Under 'Accessory' select your accessory, e.g. BNC-2090
- Upon exiting the Remote DAQ Configuration, save the new configuration.

3.4. Configuring access to the remote system

Using the Measurement & Automation Explorer, the system configuration can be protected by locking it with a password. This will

- prevent other people using the host computer or networked computers from changing the configuration
- prevent other networked host PCs from targeting LabVIEW RT to the PXI-controller unless they are included in the **RT Target: Access** list
- prevent other networked PCs from using the FTP server on the PXI-controller to write to or delete files from the hard drive of the PXI-controller

3.5. G-clamp download and installation

The G-clamp software (as contained in G-CLAMP.ZIP) consists of the following fourteen files:

- EMBEDDED.LLB
- G-CLAMP.LLB
- gNa.VI
- gKdr.VI
- gKM.VI
- gLEAK.VI
- gKA.VI
- NBFIFO.DLL
- LVANLYS.DLL
- SETTIME.DLL
- EMBEDDED.CFG
- 1PRIMARY.GTY
- 100P10_0.GT1
- 100P10_0.GT2

(1PRIMARY.GTY and 100P10_0.GT1, 100P10_0.GT2 are example template files for use with the Gsyn Threshold and the Synaptic Gain module, respectively. These patterns of virtual synaptic activity were created with the MATLAB program `Neurosim`, which can be downloaded from http://hornlab.neurobio.pitt.edu)

3.5.1. Embedded controller

- use an FTP-client program or an internet browser to establish a FTP connection between host computer and embedded controller[3]

On the embedded controller:

- create directory c:/ni-rt/g-clamp
- copy EMBEDDED.LLB into c:/ni-rt/g-clamp
- copy EMBEDDED.CFG into c:/ni-rt/g-clamp
- create directory c:/ni-rt/g-clamp/data
- create directory c:/ni-rt/g-clamp/gmodules
- copy gNa.vi, gKdr.vi, gKM.vi, gLeak.vi and gKA.vi into c:/ni-rt/g-clamp/gmodules
- create directory c:/ni-rt/g-clamp/template
- copy template-files into c:/ni-rt/g-clamp/template
- copy NBFIFO.DLL into c:/ni-rt/system
- copy LVANLYS.DLL into c:/ni-rt/system[4]
- copy SETTIME.DLL into c:/ni-rt/system[5]

To obtain the maximum loop rates, the file NI-RT.INI has to be modified:

- from the root directory of the embedded controller copy NI-RT.INI to the host computer
- on the host computer open NI-RT.INI with a text editor
- in the section 'NI-DAQ' change the key 'UseSleepingTimedNonBuffered' from TRUE (LabVIEW default) to FALSE
- save NI-RT.INI
- copy NI-RT.INI back into the root directory of the embedded controller[6]

3.5.2. Host computer

- start LabVIEW and target it to the embedded controller
- open any existing VI or make new VI
- select Tools > Network: xxx.xxx.xxx.xxx Options… (where xxx.xxx.xxx.xxx is the IP address of the embedded controller)
- in the Options dialog box select 'VI Server: Configuration'
- enable TCP/IP protocol and set the port to 3363 (these settings are LabVIEW default and should be set already)
- in 'Server resources' check all boxes to allow all
- next select 'VI Server: TCP/IP Access'
- enter the IP address of the host, allow access for this address and also check 'Strict checking'

[3] If access to the PXI controller is restricted (see section 3.4.), log-in with the correct user name and password is required.
[4] This DLL is required by 'Basic Function Generator.vi' which is part of the V-clamp module.
[5] This DLL is required to set date and time on the PXI target.
[6] Changes in NI-RT.INI become effective only after a re-start of the embedded controller. If a re-start of the embedded controller is done during a G-clamp session, G-clamp also has to be terminated and restarted.

- exit the Options dialog box by pressing 'OK'
- close the VI and switch the execution target to 'LabVIEW for Windows'
- create a directory G-clamp on the host computer and copy G-clamp.llb into this directory
- create a shortcut to G-clamp.llb and place the shortcut on the desktop or in the Start menu

3.6. Upgrading G-clamp to run under LabVIEW 7

National Instruments recently upgraded LabVIEW from version 6 to version 7. The following instructions describing installation of G-clamp under LabVIEW 7 were developed and kindly provided by Dieter Jaeger (Emory University, Atlanta, GA)

- In G-clamp.llb, replace the following LV 6.1 VIs with corresponding LV 7.0 VIs from vi.lib folder in C:\Program Files\National Instruments\LabVIEW 7.0 (LV 7.0 install location)

Utility\error.llb\Error Code Database.vi
Utility\error.llb\General Error Handler.vi
Utility\error.llb\Simple Error Handler.vi
Utility\error.llb\Find First Error.vi
analysis\baseanly.llb\Mean.vi
Utility\file.llb\Write to SGL File.vi
Utility\file.llb\Open/Create/Replace File.vi
Utility\file.llb\Close File+.vi
Utility\queue.llb\(all files)
Waveform\WDTFileIO.llb\Close WDT Array Dlog File+

- In Embedded.llb, replace the following LV 6.1 VIs with corresponding LV 7.0 VIs from vi.lib folder in C:\Program Files\National Instruments\LabVIEW 7.0 (LV 7.0 install location)

analysis\1siggen.llb\Sawtooth Wave.vi
analysis\1siggen.llb\Square Wave.vi
analysis\1siggen.llb\Triangle Wave.vi
analysis\1siggen.llb\Sine Wave.vi
measure\masignal.llb\Basic Function Generator.vi
Utility\error.llb\Error Code Database.vi
Utility\error.llb\General Error Handler.vi
Utility\error.llb\Simple Error Handler.vi
Utility\config.llb\(all files)
Utility\file.llb\Open/Create/Replace File.vi

- Mass-compile both libraries using Tools > Advanced > Mass Compile. There should be no errors found. There will be one Bad VI in Embedded.llb called SetTimeDate.vi.

Ignore this error because this VI should run properly on the PXI controller due to the availability of the settime.dll file in the "c:\ni-rt\system" folder.

- G-clamp.llb/G-clamp.vi dynamically (using an Open VI Reference) calls

G-clamp.llb/TargetList.vi
G-clamp.llb/AnalogInOutConfig.vi
Embedded.llb/List Templates.vi
Embedded.llb/SetTimeDate.vi
Embedded.llb/IV Relation.vi
Embedded.llb/Gsyn Threshold.vi
Embedded.llb/Synaptic Gain.vi
Embedded.llb/V-clamp.vi
Embedded.llb/SaveConfig.vi

- Open TargetList.vi and AnalogInOutConfig.vi, and save the recompiled versions back into G-clamp.llb.

- Open each VI in Embedded.llb library listed above and use File -> Save with Options... to save and recompile the VI into a separate library. Select Development Distribution option on left side and choose to save "To new location - single prompt" in drop down menu. Also, make sure all options available under Save entire hierarchy are selected. The radio button will shift to Custom Save on left side. Click Save. *(Should end up with 7 different libraries, List Templates.llb, SetTimeDate.llb, etc.).*

- *The above step will add any additional system files and VIs that are needed by the above individual VIs. Now, merge all the contents in the 7 library files with Embedded.llb. This can be done by copying all files in each library and pasting them into the Embedded.llb library*[7]*. Since Windows will ask whether to overwrite existing files, click Yes each time.*

  [High-lighting with Italics by Paul Kullmann: Saving the 7 VIs into different libraries and then copying the content of these libraries back into Embedded.llb seems to me an unnecessary complication. Instead save "To new location - single prompt" can be used to save the recompiled VIs (including their entire hierarchies) right back into Embedded.llb. This would also avoid the problem described in the footnote below.]

- Mass-compile Embedded.llb again to restore dependencies and download to PXI controller.

[7] Apparently starting with LabVIEW 7.0 libraries can be double-clicked and opened with the Windows Explorer. Using the Windows Explorer to copy the files from the libraries into Embedded.llb does not save all files into Embedded.llb (personal communication from Iris Oren in Ole Paulsen's lab in Oxford, UK). Instead the VI Library Manager under the Tools menu should be used.

- Download the new Embedded.llb library to the PXI controller.

3.7. Running G-clamp for the first time

- start LabVIEW and target it to the host computer
- start G-clamp (section 4.1.): Because G-clamp has not yet information about the PXI controller, it will report an error ('network address is ill-formed' and 'application reference is invalid'). Once the Status indicator reports 'Idle…', proceed with the next steps.
- select File > Preferences > PXI Target (section 4.2.3.) to provide G-clamp with the IP address of your PXI controller, the speed of its CPU and the device # of the DAQ board
- select File > Preferences > DAQ Board Configuration (section 4.2.4.) to set the analog input and out channels to be used by G-clamp and the scaling factors for each channel

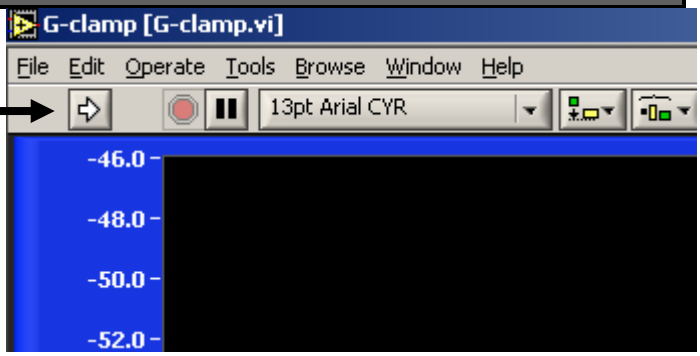
4. Using G-clamp

4.1. Starting G-clamp

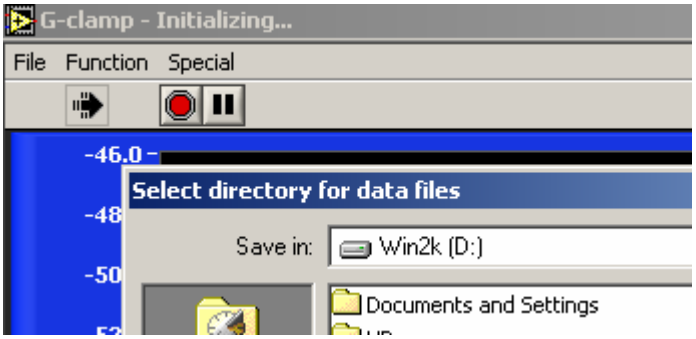
Use the G-clamp shortcut or double-click on G-clamp.llb. This will first start LabVIEW and then load the G-clamp program.

> To run G-clamp, LabVIEW has to be targeted to the host computer and not to the PXI controller!

To start the G-clamp program, press the Run-button:



The standard LabVIEW menu now changes to the G-clamp menu and a dialog box opens, prompting the user to specify a data directory on the host computer.[8] After the directory has been specified, G-clamp runs through an initialization sequence:



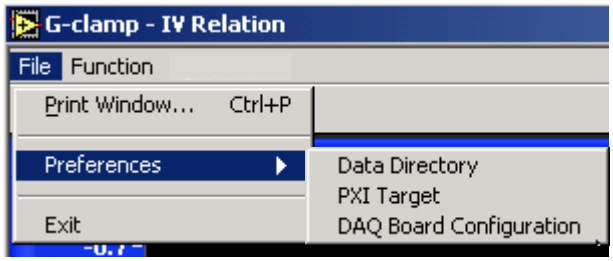
- it retrieves information (IP address, processor speed) about the embedded controller that has been used last
- it determines what template files (*.GTH and *.GT1) are available in the directory c:/ni-rt/g-clamp/template of the embedded controller

[8] Not all data gathered by G-clamp are saved to file on the host computer: The experiment module Synaptic Gain can produce large files consisting of voltage and current traces. These are saved on the embedded PC.

- it determines what virtual conductance modules are implemented
- it retrieves the last settings used for the analog input and output channels of the data acquisition board
- it synchronizes date and time of the embedded controller to the date and time on the host computer

> Do not copy new template files into c:/ni-rt/g-clamp/template or delete exisiting template files after G-clamp has been started. This corrupts the list built when G-clamp was started and will result in execution of the wrong template file. If you need to change the contents of that folder, stop and re-start G-clamp so that G-clamp can update the list.

4.2. The G-clamp menu



4.2.1. File > Print Window…

Prints the G-clamp user interface window as it is at that time on the monitor screen. Printing can also be initiated using the keyboard shortcut Crtl + P. Printing is preceded by the usual Windows print dialog.

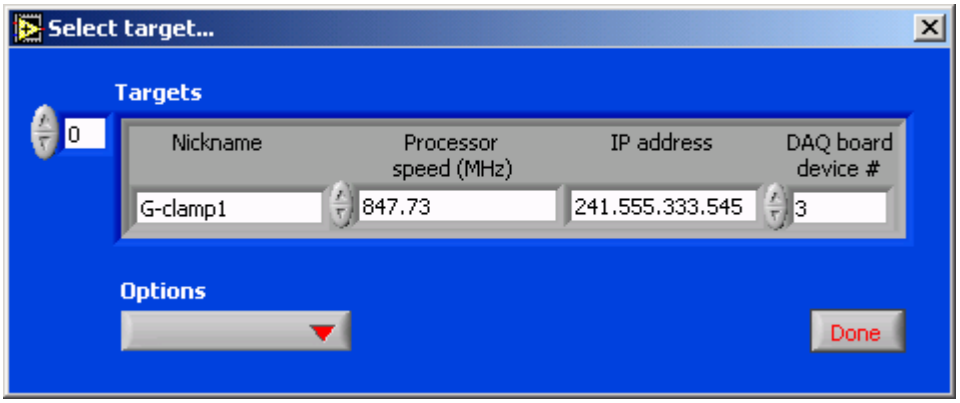
4.2.2. File > Preferences > Data Directory

Opens a dialog that allows changing the data directory on the host computer.

4.2.3. File > Preferences > PXI Target

Opens a dialog for specification of the PXI target. This is useful if the same host computer is used to control more than one PXI target. The index control at the left scrolls through the list of saved PXI targets. A PXI target can be added to or deleted from the list with the **Options** control. Upon exiting the dialog with **Done**,

G-clamp re-initializes (see section 4.1.) using the PXI target visible in the dialog. The selected PXI target will also be the one used at the next re-start of G-clamp.



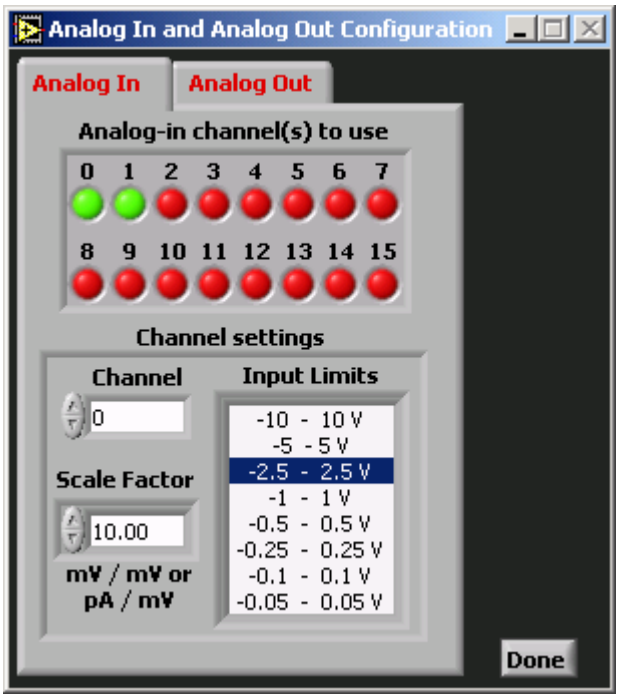
The value entered for the processor speed is used in the sub-function that determines performance of the G-clamp feedback loop (see RT Loop Info indicator in section 4.3.3.). For accurate conversion of the timing information into units of microseconds or hertz, the value entered here should be as close as possible. For example, for a PXI target with a 1.26 GHz processor the value entered here should be 1.26 x 1028 MHz = 1295.28 MHz.

The correct DAQ board device # can be determined using the Measurement & Automation Explorer (MAX).

4.2.4. File > Preferences > DAQ Board Configuration

Opens a window for configuration of the analog input and output channels of the data acquisition board. Once configuration is complete, the settings are made permanent by writing them to a configuration file from which they are loaded whenever G-clamp is started.

Analog In: The round buttons activate (green) or deactivate (red) analog input channels. Version 1 of G-clamp uses two AI channels: The channel with the

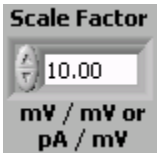


lowest number is used for acquiring the membrane potential and the channel with the next higher number is used for acquiring the current injected via the recording electrode. Any combination of two out of the 16 AI channels[9] of the E-series data acquisition board can be used. If more than two channels are activated, G-clamp will perform the data acquisition on these additional channels, but these data will not be saved or used in any way.

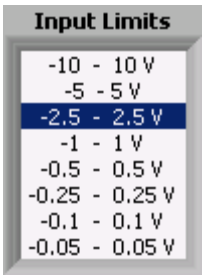
For each analog input channel two options need to be specified: the scale factor and the input limits. To select a channel, use the 

control. Upon selecting a channel, the scale factor and input limits will be updated to display the current settings for the selected channel.

The scale factor depends on the outputs of the recording amplifier. For example, the Axoclamp-2B amplifier (Axon Instruments) internally amplifies the recorded membrane potential by a factor of 10 before it is made available at the output. For the recorded current, the output of the Axoclamp-2B amplifier depends on the gain of the headstage, necessitating adjusting the scale factor when the headstage is changed.

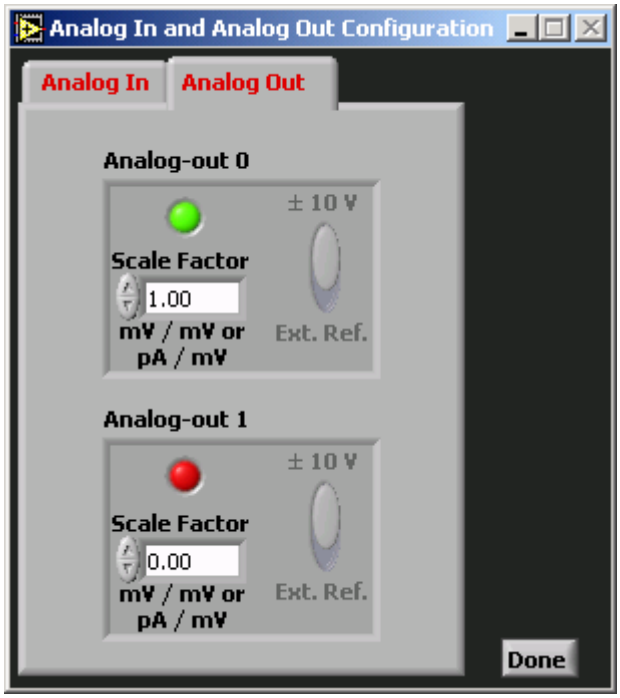


[9] This assumes that the analog input channels have been set up for single-ended use and not for differential use.



Input limits[10] allow adjusting the gain of the data acquisition board to the dynamic range encountered. This is done for each analog input channel individually. Thus, with the Axoclamp-2B amplifier an input limit of ±2.5 V for the channel acquiring the membrane potential gives a measuring range of ±250 mV. For the channel acquiring the current injected by the amplifier, an input limit of ±1 V covers the ±10 nA range of current that can be injected by the HS-2A headstage (gain of the headstage: 0.1; G-clamp scale factor: 0.1).

Analog Out: Similar to Analog In, the round buttons activate and deactivate an analog output channel. As version 1 of G-clamp uses only one AO channel (connected to the current command of the recording amplifier), activating one channel automatically de-activates the other one.



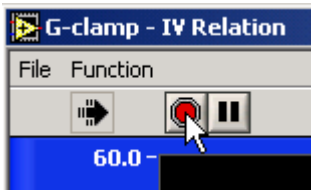
Scale factor depends on the sensitivity of the current command input of the recording amplifier, which in turn can depend on the gain of the headstage used.

The ±10 V/Ext.Ref. control is intended for use in future versions of G-clamp and disabled in version 1.

[10] The list of input limits implemented in G-clamp reflects the possible gain settings for the 6052E DAQ board from National Instruments. If you have another board, consult the board's manual for its gain options and the G-clamp Programmers Guide about how to adjust the Input Limits control to a different set of gain settings.

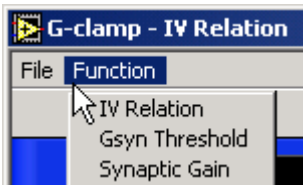
Exits the analog input and output configuration and saves the new settings. After the settings have been written to a configuration file on the embedded controller, the configuration window closes and G-clamp is ready to use the new settings.

4.2.5. File > Exit



Choosing Exit from the File menu stops G-clamp after any ongoing process has finished and writes the current settings for the virtual conductances (used/not used and g-value) to a configuration file from where they are loaded the next time G-clamp is started. Alternatively, using the LabVIEW stop button kills any ongoing process and stops G-clamp immediately without saving the virtual conductance settings. With both methods the G-clamp user interface window stays open after G-clamp has stopped.

4.2.6. Function > IV Relation; Gsyn Threshold; Synaptic Gain



Selects an experiment module and activates the controls and indicators specific for the selected module. Controls and indicators specific to the old module are deactivated. The currently active module is displayed in the G-clamp window title. By default G-clamp starts with the IV Relation module.

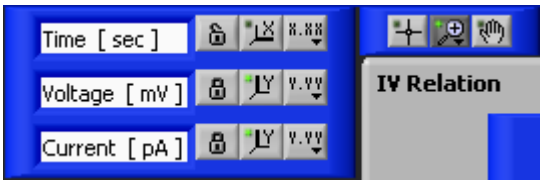
The three experiment modules (IV Relation; Gsyn Threshold; Synaptic Gain) presently available in G-clamp are explained in detail in section 5.

4.3. Controls and indicators on the G-clamp user interface

The G-clamp user interface contains four groups of controls and indicators:

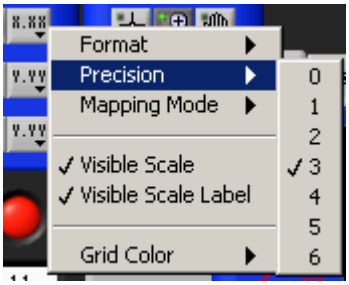
1 Voltage and current trace display-

Each signal has its own y-axis scale: membrane voltage is scaled on the left y-axis; membrane current on the right. Standard LabVIEW controls for navigating and customizing the display are located below the graph.



The lock-button toggles between auto-scaling and manual scaling of the corresponding axis. With auto-scaling, the axis will be re-scaled whenever a new trace is added

to update displayed graph. With manual scaling the axis can be set to any range by editing its end values.[11]



This button provides formatting options for an axis. For example, the option Precision can be used to change the number of decimal digits in axis labels. The option Grid Color can be used to add a grid to the graph. Note that changes of these formatting options become only permanent if G-clamp is saved before it is closed.[12]

This standard LabVIEW button toggles the mouse into cursor mode. The button is functionless here, as this graph does not have cursors.

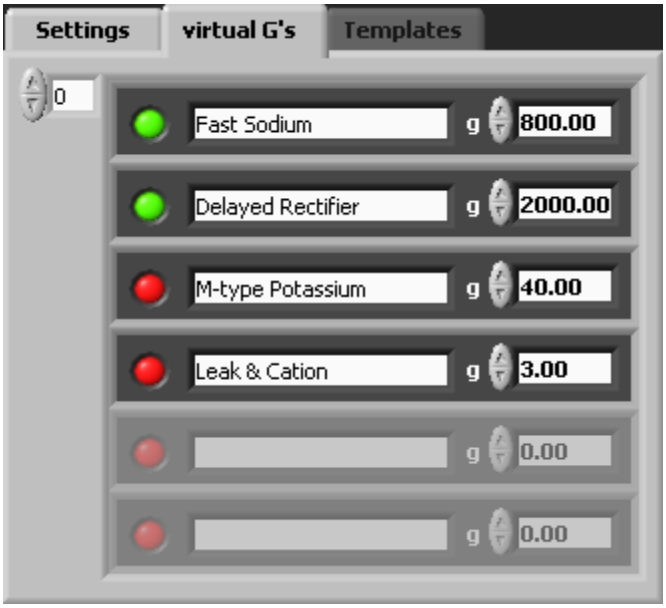
The zoom button offers several options for zooming into the graph and for undoing the last zoom action.

The hand tool can be used to grab the graph and move it around in the display area.

2 Experiment controls-

This tab-controlled window contains three pages with all of the parameters needed to set up any of the experiment modules. Depending on the experiment module chosen via the Function-menu, the exact content of the tab-control pages changes by de-activating unnecessary controls and activating required controls. This is especially prominent in the **Settings**-page. Controls on the **Settings**-page are therefore explained in the context of individual experiment modules (sect. 5).

[11] Some of the experiment modules auto-scale the axes programmatically at the end of a series of data acquisitions, thus over-riding the setting of this control.
[12] Stop G-clamp (see 4.2.2.). This brings back the standard LabVIEW menu (see 4.1. top figure). Select File > Save to make the changes permanent.

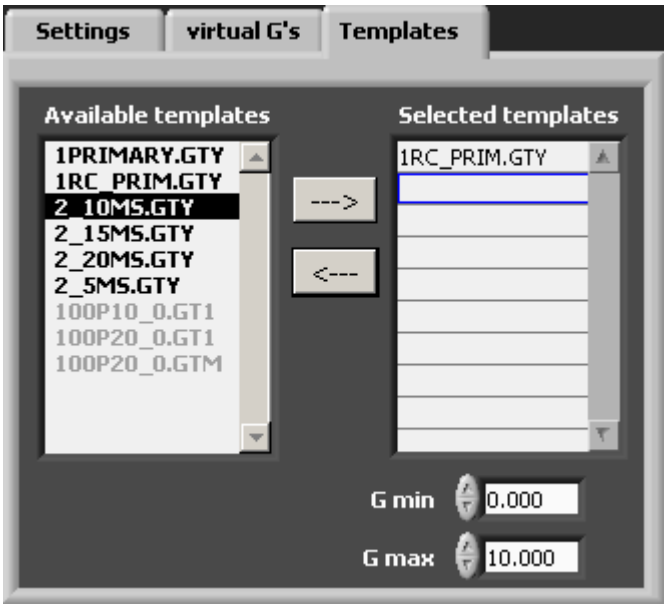


The **virtual G's**-page is used to activate/deactivate those virtual conductances that require calculation 'on the fly' and to specify their maximal values (g_{max}, also known as g-bar). To activate/deactivate a conductance press its LED:  indicates activated,  indicates deactivated. To change the value of g either use the up/down control  or directly edit the value. Changes must be made before starting an experiment.

The virtual conductances can be used in all experiment modules. The settings for the virtual conductances are global, i.e. upon switching from one experiment module to another the on/off settings for the conductances and their g-values remain in effect.

The **Templates**-page allows the user to select one or more pre-defined conductance templates. Upon program start, G-clamp checks the directory c:\ni-rt\g-clamp\template on the embedded controller for valid template files and builds a list of available templates. Availability is further narrowed down by the choice of experiment module: Thus the Gsyn Threshold-module uses only template files of type *.GTY. All other template file types cannot be used with this experiment module and are therefore grayed out, indicating that they cannot be selected. To

select a template highlight it in the left list and then press ---> to add it to the list of selected templates. Highlighting a selected template and then pressing <--- removes the template from the selection. For repetitive execution of an experiment module, a series of templates can be assembled. Once started, the experiment module will execute the top template, remove it from the list and keep executing until the list of selected templates is empty. The remaining controls on this page are experiment module-specific and therefore explained in detail in section 5.2 and 5.3. The entire Template page is unavailable for the experiment module IV Relation, because this module does not use templates.

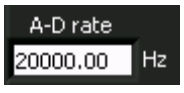


3 General controls and indicators-

Starts execution of the selected experiment module. After the module has started, the button changes to 

. However, only the modules performing a series of data acquisitions (IV Relation; Gsyn Threshold; V-clamp) can be stopped.

Indicates the number of iterations completed durings experiments that employed repeated trials (IV Relation; Gsyn Threshold).

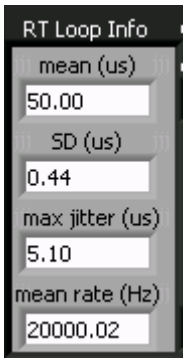


Indicates actual performance of the data acquisition board. Because the oscillator on the data acquisition board operates at a defined

frequency, not all requested sampling rates are possible. For example, with the 20 MHz oscillator on NI E-series boards, a requested sample rate of 24000 Hz results in an actual sample rate of 24009.6 Hz.

Controls whether the data of an experiment are saved or not. This control is read out after the last iteration of an experiment and can therefore be set after the experiment was started and is still in progress.

Indicates the name of the last data file created. File names are assigned automatically by G-clamp. File names consist of an 8-digit number reflecting day and time of the experiment and a 3-letter extension specific for the experiment module: ddhhmmss.??? where dd denotes day of month, hh hour (24-hour clock), mm minute, ss seconds. The 3-letter extensions ??? are IVB for the IV Relation module, GTH for the Gsyn Threshold module, GAJ and GAI for the Synaptic Gain module. For more details about data files see section 8.



The RT Loop Info indicator gives information about how well the G-clamp feedback loop maintained the intended cycle rate. The underlying function creates a time stamp for each iteration of the feedback loop[13] and uses the time stamps to calculate mean loop interval, standard deviation, largest deviation from the mean interval (max. jitter) and the mean cycle rate. Because of the memory requirement for the time stamps, the maximum number of time stamps is limited to 1,000,000. In experiments where the feedback loop operates beyond 1 million cycles, errors after that point go undetected. As a general rule: G-clamp is performing as

[13] The time stamp is created immediately after the feedback loop outputs the new current command to the recording amplifier.

intended when max jitter does not exceed one half of the intended sample interval, because this means that the system did not miss a single individual A-D conversion .[14]

4 Analysis results–

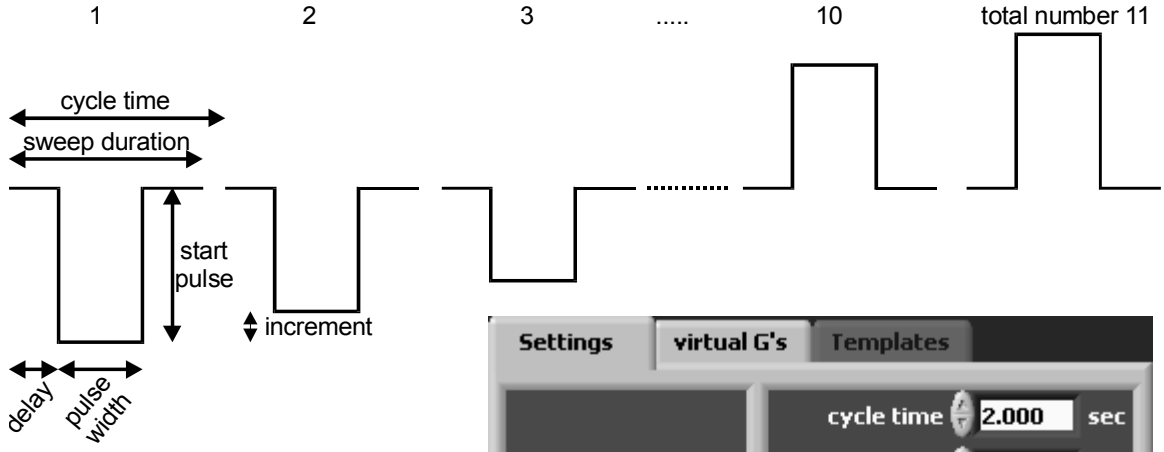
This tab-control contains on its three pages graphs and indicators for the individual experiment modules. Each page is explained in detail in the context of its experiment module (section 5).

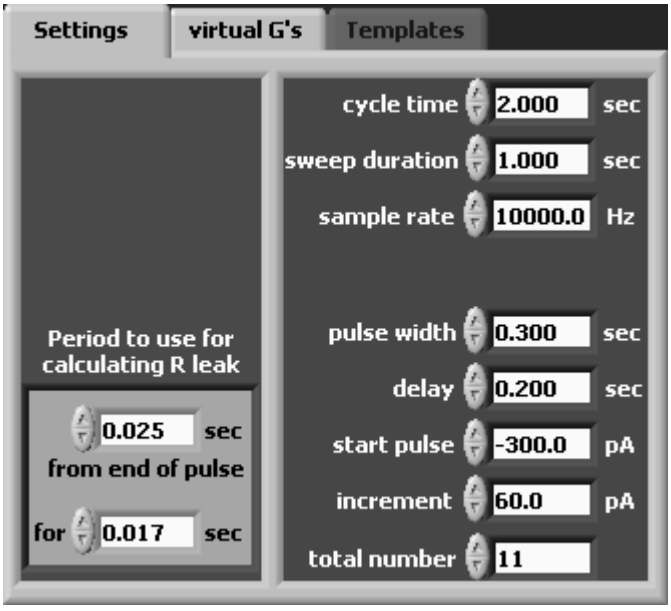
[14] For more details on the timing of events between the data acquisition board and the G-clamp feedback loop see section 9.

5. Experiment Modules

5.1. IV Relation

This module performs a series of constant current injections to measure the steady-state current-voltage relationship of a cell. The acquisition and pulse parameters are defined in the Settings-page of the experiment control:

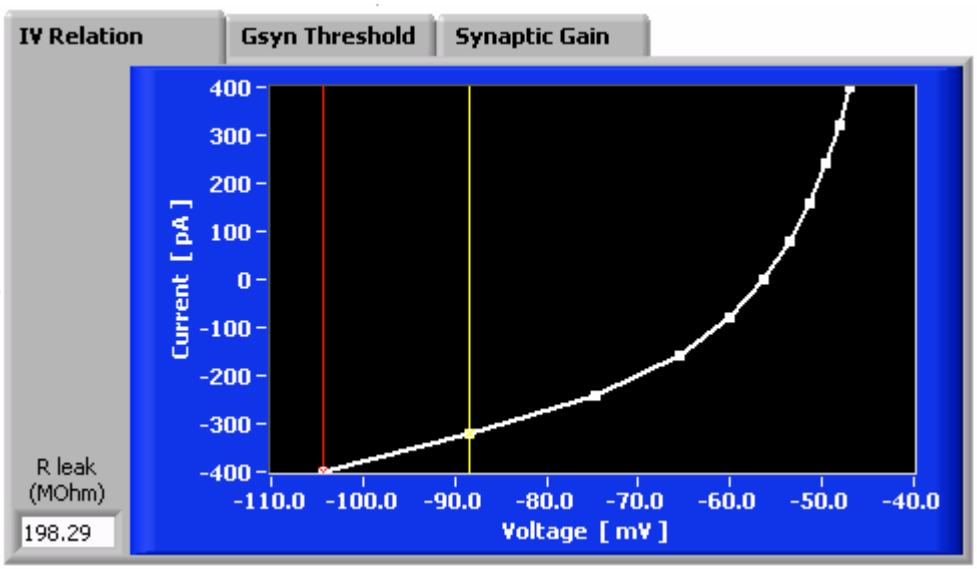




Two additional controls specify which parts of the recorded traces are used to determine the current-voltage relationship. G-clamp takes a segment of the voltage and current traces and calculates the mean value of all data points within that segment. The start point of this segment is set with 

and its length with 

. The values are then plotted and displayed the IV Relation page of the analysis results tab-control. This plot contains two cursors (red and yellow vertical lines), which can be moved the voltage axis. The two data points marked by the cursors define a straight line.



G-clamp calculates the slope of this line and displays the result in the R leak indicator to the left of the plot.

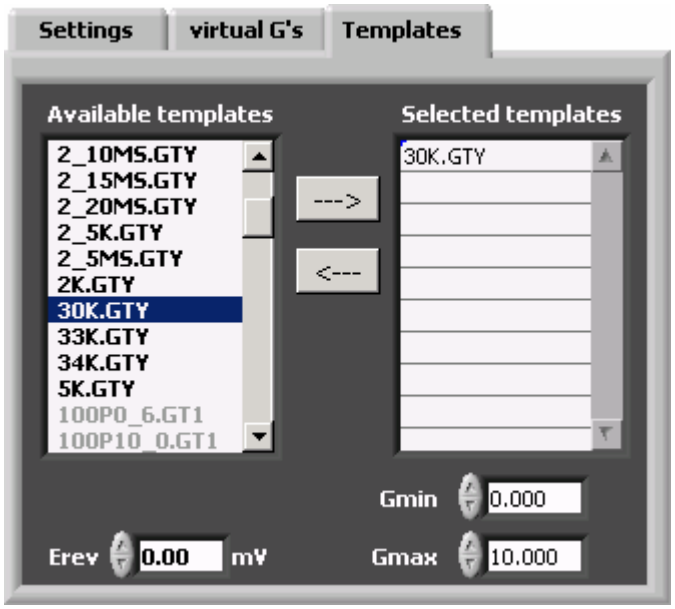
Data file: G-clamp stores both, the acquired traces (voltage and current) as well as the values calculated for the IV plot in one file, having the file name extension IVB. Save location is the directory on the host computer specified by the user when G-clamp was started. For more detailed information about the data file structure, see section 8.

It should be noted that unlike standard current-clamp systems, G-clamp allows the user to add a virtual conductance to a cell and measure its effect upon the steady-state I-V relation.

5.2. Gsyn Threshold

The purpose of this module is to determine how strong a synapse needs to be to bring the postsynaptic neuron to fire an action potential (threshold synaptic conductance). To find threshold-g_{syn} the module implements a virtual synapse from a pre-defined conductance waveform (template). It conducts an automated binary search in which synaptic strength is iteratively increased or decreased depending on whether the last setting elicited an action potential or not.

1. Use the **Templates**-page of the experiment tab control to select a template (for how-to see section 4.3.). Valid templates for the Gsyn Threshold module are templates of type *.GTY. Erev sets the reversal potential and Gmin and Gmax control the scaling of the synaptic conductance implemented by the template:



$$g_{syn} = (Gmin + Gmax) / 2$$

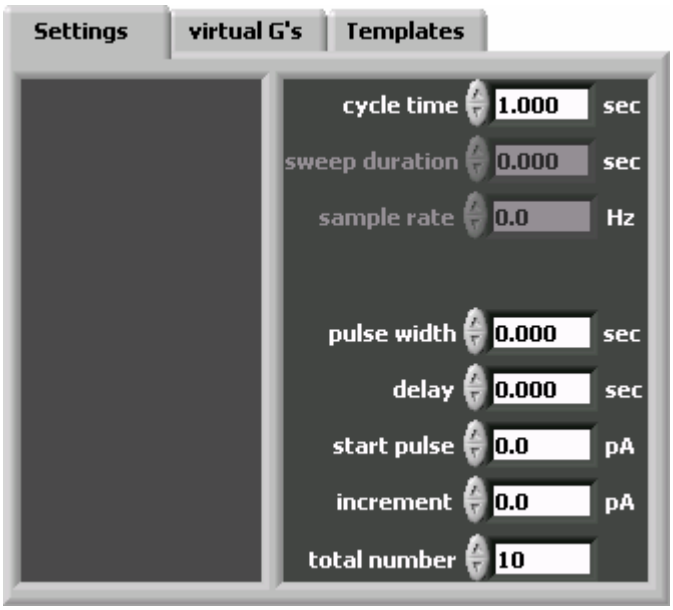
For the 1st trial (i = 1), Gmin and Gmax take on the values specified by the user. For subsequent trials i + 1, G-clamp adjusts Gmin and Gmax depending on the outcome of the preceding trial i:

case $g_{syn}(i)$ subthreshold, then $Gmin(i+1) = g_{syn}(i)$

case $g_{syn}(i)$ superthreshold, then $Gmax(i+1) = g_{syn}(i)$

This algorithm will find a stable solution within about 12 trials if Gmin < g_{syn} < Gmax. Otherwise will converge onto Gmin if g_{syn} < Gmin or Gmax if Gmax < g_{syn}. The program assumes the template has a maximum conductance of 1, which yields Gmin, Gmax and g_{syn} in units of nS.

2. The **Settings**-page of the experiment tab provides for setting the **cycle time** and **total number** parameters.

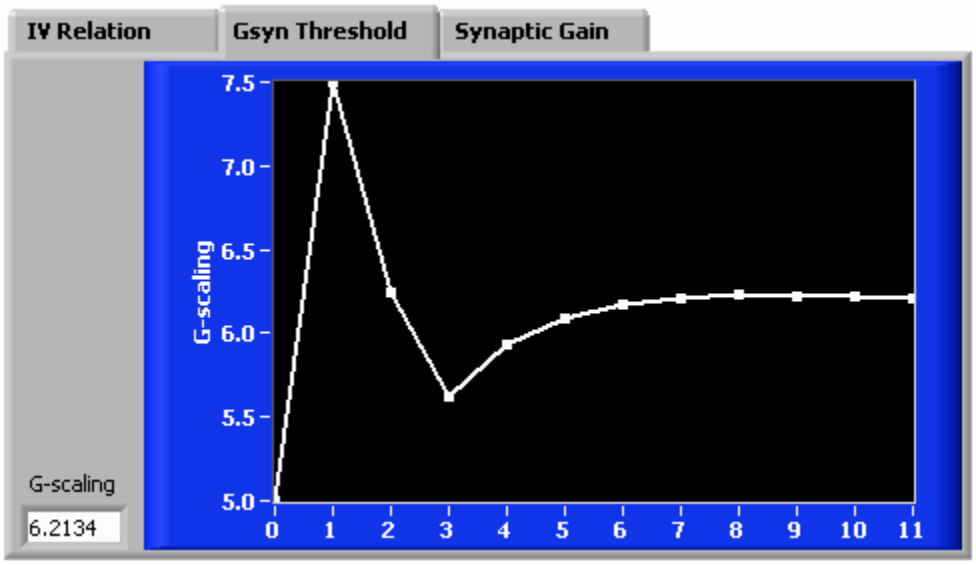


   - **cycle time** is the time from start of one trial to start of the next trial. The minimum cycle time therefore corresponds to the sweep duration. Sweep duration is determined by the length of the template file. Because G-clamp reads this information as well as the sample rate from the header of the template file, the **sweep duration** and **sample rate** controls on the **Settings**-page are not needed with this experiment module and the controls are inactivated and grayed out.

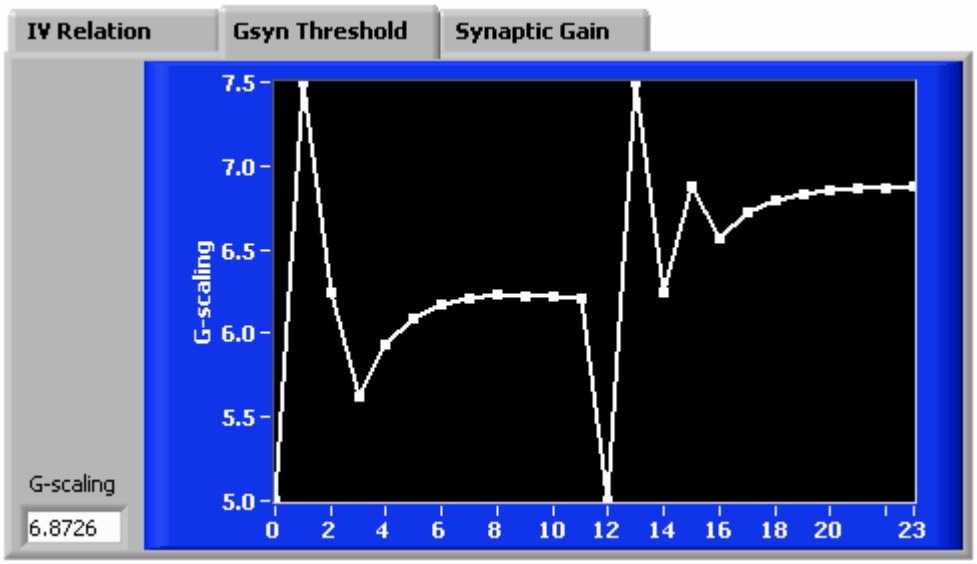
     Note that the controls **pulse width**, **delay**, **start pulse** and **increment** are <u>not</u> grayed out and inactivated. While these controls become all set to 0 when this experiment module is selected for the 1st time after starting G-clamp, they can be used to impose a current step pattern on the sweeps. The rules for using these controls in the Gsyn Threshold module are the same as in the IV Relation module (section 5.1.).

   - **total number** specifies the maximum number of trials the experiment module will use to find threshold-g_{syn}.

The progress of an experiment and its final result can be observed on the Gsyn Threshold page of the analysis results tab-control. One data point representing the g_{syn} value used during a sweep is added to the chart after the sweep has been completed. A numerical representation of that value is also given in the bottom left indicator on that page.



This chart retains its content from a previous Gsyn Threshold experiment when a new one is started and simply adds the new data points to the chart:



This feature provides an easy way to monitor whether thresh-g_{syn} is stable or whether it changes in a systematic fashion over time. The memory of the chart operates like a FIFO buffer that can store up to 100 g_{syn} values. Once the chart contains 100 values and the experiment module adds another value (to the right), the oldest value (to the left) gets pushed off the chart.

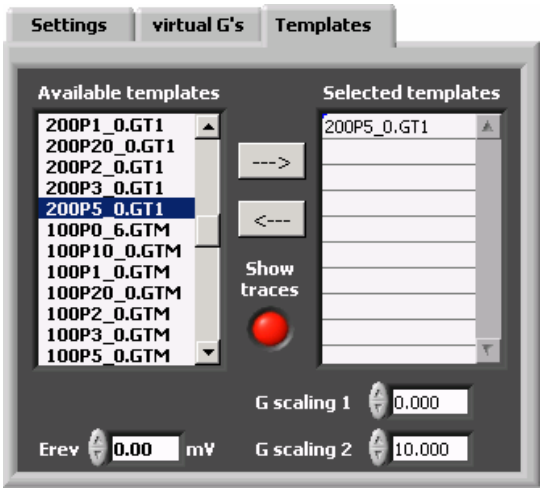
To clear the chart from all data, right click the chart and select **Clear Chart** in the upcoming menu.

5.3. Synaptic Gain

This experiment module stimulates a cell with specific patterns of synaptic activity by reading appropriate template files containing synaptic conductance waveforms. At the end of the experiment, the program automatically calculates synaptic gain by counting the number of action potentials elicited by the virtual EPSPs and dividing this number by the number of presynaptic events per synapse in the associated template file (see Wheeler, Kullmann and Horn (2004) for more details about defining synaptic gain).

In this experiment, the strength of the synapses implemented by the template file can be scaled using the results from the Gsyn Threshold experiment. By sequentially testing a series of template files reflecting different rates of presynaptic activity, the system can automatically measure and display the relation between presynaptic firing rate and synaptic gain.

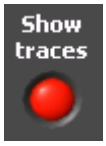
1. Use the **Templates**-page of the experiment tab control to select a template (for how-to see section 4.3.). Selectable templates for this module are templates of type *.GT1.

   The **G scaling 1** and **G scaling 2** control act to scale the conductance waveforms in the *.GT1-file (**G scaling 1**) and in its matching *.GT2-file (**G scaling 2**). Because each *.GT1-file has exactly one

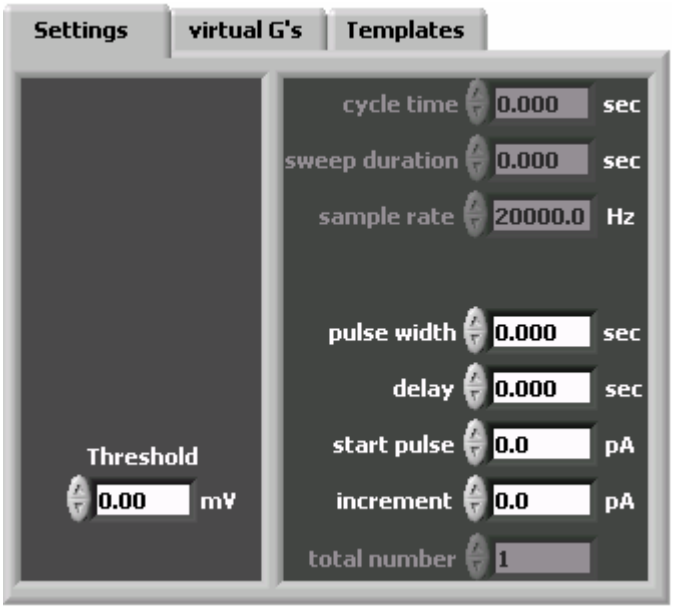
matching *.GT2-file[15] and because selection of a *.GT1-file from the list of available templates implicitly results in concurrent selection of the matching *.GT2-file, the *.GT2-files do not show up in the list of available or selected templates.

Determines the reversal potential of the conductance implemented by the selected template. Applies also to the matching *.GT2-file if the selected template is of type *.GT1 (see above paragraph).



For long template files the transfer of the acquired traces from the embedded controller to the host computer for display can take seconds up to minutes. To allow fast continuation with the next experiment, this control per default disables data transfer and display of the traces. (Although short in comparison, a delay is still noticeable if the acquired traces are saved to file on the embedded controller.) This control is read out at the end of a template's execution.

2. The **Settings**-page of the experiment tab contains the **Threshold** parameter. By setting this analysis parameter, one chooses the membrane voltage which must be crossed in order to be detect and count



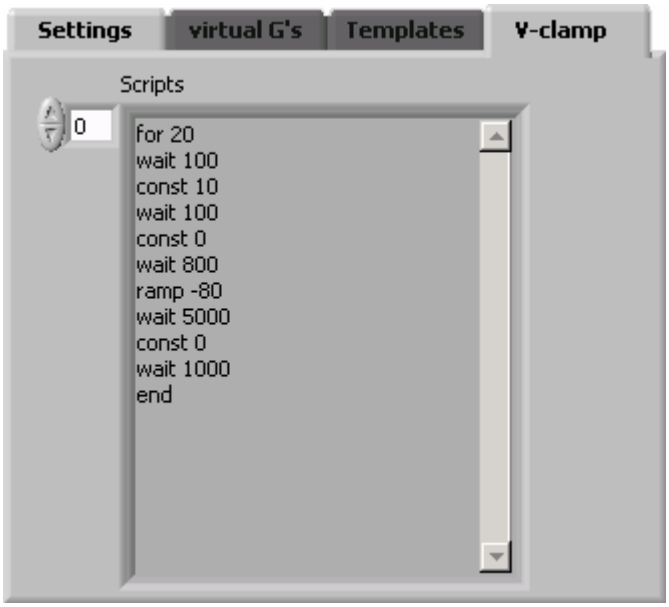
[15] Matching *.GT1- and *.GT2-files have the same filename, e.g. 200P5_0.GT1 and 200P5_0.GT2.

postsynaptic action potentials used in the measurement of synaptic gain..

Note also that in this the gain experiment, **cycle time** has been disabled and grayed out. This is because template files contain a parameter "wait after execution [sec]" which guarantees a minimum recovery period before execution of the next template file starts. If the data of the experiment are saved and especially after long template files this waiting period might well be outlasted by the time required to write the data to disk.

5.4. V-clamp

The V-clamp module is intended for voltage clamp experiments. It is a new module added to G-clamp since publication of our J. Neurophysiol. paper in January 2004. Similar to the IV Relation module, it outputs a command signal to the voltage command input of the recording amplifier. The command signal can be a (series of) square pulse(s) or ramp(s) or a combination of these. The controls that are used to define the command signal with the IV Relation module (on the 'Settings' page) allow only one square pulse to be defined. Instead of adding more controls to specify additional command changes, the V-clamp module uses a scripting protocol to define the command signal.

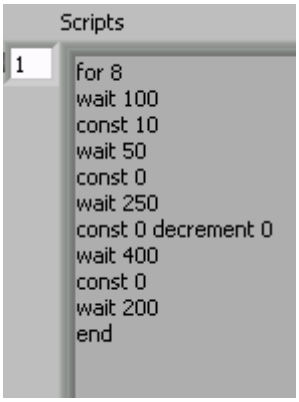


Available commands:

- **for** – The beginning of a loop. Currently the only loop allowed (and required) is looping through the whole script.
- **const** – Changes the output in a step-like manner to the value after the command. The command value can be incremented or decremented in successive iterations of the script.

- **ramp** – Changes the output in a ramp-like manner to the value after the command. The speed of the ramp is determined by the next wait-command. Like for the const-command, the command value can be incremented or decremented in successive iterations of the script.
- **wait** – Determines the duration for an output command (const) or how long it takes before the command value is reached (ramp).

- **increment** – Increments the command value of the preceding ramp- or const-command by the value following increment each time the script iterates. The command follows the const or ramp command in the same line, separated by a space from the value of the const/ramp-command.



- **decrement** – Decrements the command value of the preceding ramp- or const-command by the value following decrement each time the script iterates. The command follows the const or ramp command in the same line, separated by a space from the value of the const/ramp-command.

Script syntax:

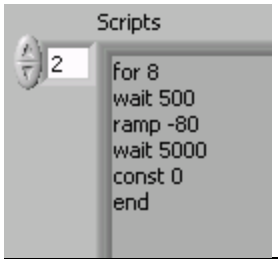
- Each command (except the *end*-command) is followed by a numerical value.
- Command and numerical value have to be separated by a space.
- The 1st command has to be the *for*-command even if the script is a single data acquisition

- The 2nd command has to be a *wait*-command.
- Each command that changes the output (*const*, *ramp*) is followed by a wait command in the next line.
- Time unit is ms and voltage unit is mV.

To see the scripts that are available, use the index control at the top left corner of the 'Scripts' indicator. Scripts can be modified by editing them in the 'Scripts' indicator and new scripts can be added by using the index control to scroll to the end of the script list and the first free field. To make modifications or additions permanent, stop G-clamp and right-click while pointing on the index control. This brings up a menu. Select Data Operations > Make Current Value Default[17] and subsequently save G-clamp.vi.

Note that the duration of a data acquisition, the interval rate of successive iterations and the sampling rate are still set with the controls 'sweep duration', 'cycle time' and 'sample rate' on the 'Settings' page of the Tab-control. Thus if 'sweep duration' is shorter than the script (the sum of all wait-commands), data acquisition and command updating will stop after the time set by 'sweep duration'.

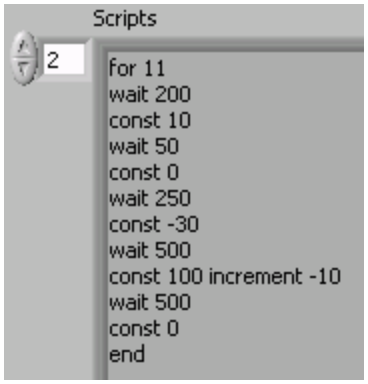
Examples:



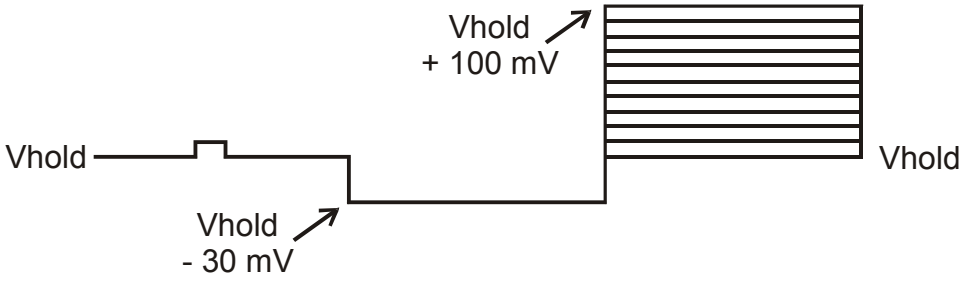
This script, after an initial delay of 500 ms, executes a ramp that hyperpolarizes the cell within 5000 ms by 80 mV from the current

[17] If you right-click while pointing into the script text field, the upcoming menu also allows to delete or insert individual fields into the list.

holding potential before returning to the original holding potential. This pattern is repeated 8 times.



This script combines a small, brief voltage step (to monitor series resistance) with a series of successively smaller depolarizing voltage steps (to activate a membrane conductance) that are preceded by a constant hyperpolarizing voltage step (to remove inactivation of the conductance).



6. Virtual Conductances

G-clamp v1.1 contains 5 virtual conductance modules implementing

- a fast Na^+-conductance (E_{rev} +60 mV)
- a delayed rectifier K^+-conductance (E_{rev} -90 mV)
- a muscarinic (M-type) K^+-conductance (E_{rev} -90 mV)
- a linear leak conductance (E_{rev} -40 mV) and
- a transient (A-type) K^+-conductance (E_{rev} -84 mV).

The conductance values are calculated in real time using the Euler method to integrate the ordinary differential equations that describe them. The equations describing the current kinetics of the first 4 conductances are based on the kinetics of these currents in bullfrog sympathetic B neurons (Yamada et al., *Methods in Neuronal Modeling: From Synapses to Networks,* Eds: C. Koch and I. Segev. Cambridge, MA: MIT Press, 1989, p. 97–133). Since the inactivation of the delayed-rectifying potassium current is slow, we have omitted the inactivation parameter from our model. The effects of including inactivation have been described elsewhere (Schobesberger et al., J.Neurophysiol. 83:1912-1923, 2000). The equations describing the current kinetics of the transient (A-type) K^+-conductance are based on recordings from rat dopaminergic neurons (Hahn et al., J. Neurosci. 26:10859-10866, 2003).

Fast inactivating sodium current

$$I_{Na}(V, t) = \bar{g}_{Na}\, m^2(V, t)\, h(V, t)\, (V - E_{Na})$$

$$dm(V, t)/dt = [m_\infty(V) - m(V, t)] / \tau_m(V) \qquad \text{(activation variable)}$$

$$m_\infty(V) = \alpha_m(V) / [\alpha_m(V) + \beta_m(V)] \qquad \text{(steady-state)}$$

$$\tau_m(V) = 2 / [\alpha_m(V) + \beta_m(V)] \quad \text{(time constant)}$$

$$\alpha_m(V) = [0.36\ (V + 33)] / [1 - \exp\{-(V + 33) / 3\}] \quad \text{(forward rate)}$$

$$\beta_m(V) = [-0.4\ (V + 42)] / [1 - \exp\{(V + 42) / 20\}] \quad \text{(backward rate)}$$

$$m(V, t+\Delta t) = m(V, t) + [[m_\infty(V) - m(V, t)] / \tau_m(V)]\ \Delta t \quad \text{(Euler integration)}$$

$$m(V, t+\Delta t) = m_\infty(V) + [m(V, t) + m_\infty(V)] \exp\{-\Delta t / \tau_m(V)\} \quad \text{(exponential Euler)}$$

$$dh(V, t)/dt = [h_\infty(V) - h(V, t)] / \tau_h(V) \quad \text{(inactivation variable)}$$

$$h_\infty(V) = \alpha_h(V) / [\alpha_h(V) + \beta_h(V)]$$

$$\tau_h(V) = 2 / [\alpha_h(V) + \beta_h(V)]$$

$$\alpha_h(V) = [-0.1\ (V + 55)] / [1 - \exp\{(V + 55) / 6\}]$$

$$\beta_h(V) = 4.5 / [1 + \exp\{-V / 10\}]$$

$$h(V, t+\Delta t) = h(V, t) + [[h_\infty(V) - h(V, t)] / \tau_h(V)]\ \Delta t$$

$$h(V, t+\Delta t) = h_\infty(V) + [h(V, t) + h_\infty(V)] \exp\{-\Delta t / \tau_h(V)\}$$

Delayed-rectifier potassium current

$$I_K(V, t) = \bar{g}_K\, n^2(V, t)\ (V - E_K)$$

$$dn(V, t)/dt = [n_\infty(V) - n(V, t)] / \tau_n(V) \quad \text{(activation variable)}$$

$$n_\infty(V - 20) = \alpha_n(V - 20) / [\alpha_n(V - 20) + \beta_n(V - 20)]$$

$$\tau_n(V) = 1 / [\alpha_n(V) + \beta_n(V)]$$

$$\alpha_n(V) = [0.0047\ (V + 12)] / [1 - \exp\{-(V + 12) / 12\}]$$

$$\beta_n(V) = \exp\{-(V + 147) / 30\}$$

$$n(V, t+\Delta t) = n(V, t) + [[n_\infty(V) - n(V, t)] / \tau_n(V)] \, \Delta t$$

$$n(V, t+\Delta t) = n_\infty(V) + [n(V, t) + n_\infty(V)] \exp\{-\Delta t / \tau_n(V)\}$$

M-type potassium current

$$I_M(V, t) = \bar{g}_M \, w(V, t) \, (V - E_M)$$

$$dw(V, t)/dt = [w_\infty(V) - w(V, t)] / \tau_w(V) \qquad \text{(activation variable)}$$

$$w_\infty(V) = 1 / [1 + \exp\{-(V + 35) / 10\}]$$

$$\tau_w(V) = 1000 / [3.3 \, [\exp\{(V + 35) / 40\} + \exp\{-(V + 35) / 20\}]]$$

$$w(V, t+\Delta t) = w(V, t) + [[w_\infty(V) - w(V, t)] / \tau_w(V)] \, \Delta t$$

$$w(V, t+\Delta t) = w_\infty(V) + [w(V, t) + w_\infty(V)] \exp\{-\Delta t / \tau_w(V)\}$$

The implemented leak conductance has two components:

a) Cyclic nucleotide-gated cation leak current

$$I_{CNG}(V) = g_{CNG} \, (V - E_{CNG})$$

b) Background leak current

$$I_{leak}(V) = g_{leak} \, (V - E_{leak})$$

Transient (A-type) potassium current

$$I_A(V, t) = \bar{g}_A \, m^3(V, t) \, h(V, t) \, (V - E_K)$$

$$dm(V, t)/dt = [m_\infty(V) - m(V, t)] / \tau_m(V) \qquad \text{(activation variable)}$$

$$m_\infty(V) = 1 / [1 + \exp\{-(V + 24.8) / 13.9\}] \qquad \text{(steady-state)}$$

$$\tau_m(V) = 2 - (1.6 / [1 + \exp\{-(V + 20) / 15\}]) \quad \text{time constant)}$$

$$m(V, t+\Delta t) = m(V, t) + [[m_\infty(V) - m(V, t)] / \tau_m(V)] \, \Delta t \quad \text{(Euler integration)}$$

$$m(V, t+\Delta t) = m_\infty(V) + [m(V, t) + m_\infty(V)] \exp\{-\Delta t / \tau_m(V)\} \quad \text{(exponential Euler)}$$

$$dh(V, t)/dt = [h_\infty(V) - h(V, t)] / \tau_h(V) \quad \text{(inactivation variable)}$$

$$h_\infty(V) = 1 / [1 + \exp\{(V + 78.7) / 9.2\}]$$

$$\tau_h(V) = 28 - (9.4 / [1 + \exp\{-(V - 2) / 16\}])$$

$$h(V, t+\Delta t) = h(V, t) + [[h_\infty(V) - h(V, t)] / \tau_h(V)] \, \Delta t$$

$$h(V, t+\Delta t) = h_\infty(V) + [h(V, t) + h_\infty(V)] \exp\{-\Delta t / \tau_h(V)\}$$

Voltage-independent synaptic conductances are implemented by reading a pre-determined conductance waveform (see section 7. Template Files) from file.

7. Template Files

Template files contain pre-defined conductance waveforms. Currently G-clamp uses template files to implement virtual synapses in two of its experiment modules:

- the Gsyn Threshold module uses template files ending with GTY
- and the Synaptic Gain module uses template files ending with GT1 and GT2.
- All template files are binary files containing single precision floating point numbers (SGL = 4 bytes/number) and have the same basic structure, i.e. a header containing information about the template, followed by the template. The length of the header differs for the different file types. *.GTY template files have 19 fields (single precision floating point numbers, SGLs) to store information. To make the disk read operation of the long template files used with the Synaptic Gain module (*.GT1, *.GT2) more efficient (the real-time OS of the embedded controller performs hard disk I/O in blocks of 512 bytes), header length for *.GT1 and *.GT2 files has been extended to 128 fields. Not all of the information contained in the header is actually necessary for using the template file in G-clamp. Given below is the information we store in the first 14 header fields. The numbers at the end of some of the fields indicate the experiment module for which content of this field is required (Gsyn Threshold module - 1 or Synaptic Gain module - 2).

  1. time interval between consecutive points [ms] 1,2
  2. threshold-gsyn [nS]
  3. gsyn-bar for primaries [nS]
  4. gsyn-bar for secondaries [nS]
  5. initial settling time [ms]
  6. amplitude of fpre modulation [Hz]
  7. rate of fpre modulation [Hz]

8. phase of fpre modulation for primaries [rad]
9. phase of fpre modulation for secondaries [rad]
10. number of secondaries[18] 2
11. fpre [Hz][19] 2
12. length of template [ms] 2
13. not used
14. wait after execution [sec] 2

15 and >: not used

File names have to be in DOS 8.3 notation to be FTPable.

8. Data Files

8.1. Names

Data file names have the general form *ddhhmmss**** in which

- *dd* is the day of month,
- *hh* is the hour (24-hour clock),
- *mm* is the minute,
- *ss* is the second when the experiment finished, and
- *** indicates the experiment type (IV Relation: IVB; Gsyn Threshold: GTH; Synaptic Gain: GAI and GAJ; V-clamp: PKP)
  GAJ: can contain data from a series of templates: *ddhhmmss* reflects end of first experiment in the series)

8.2. Content

Each G-clamp experiment module has its own data file format.

[18] Whatever its actual value, this field is without meaning in *.GT1 files.
[19] fpre is the frequency of *all* events. For a synaptic gain experiment using a pair of *.GT1 and *.GT2 files, the total number of synaptic events in the combined template is thus:
($fpre_{GT1}$ [Hz] + $fpre_{GT2}$ [Hz]) * (length of template [ms] / 1000))

8.2.1. IV Relation - *.IVB

- a single precision floating point number (SGL – 4 bytes) giving the number of iterations done
- a SGL giving the amplitude of the start (1^{st}) pulse
- a SGL giving the increment for subsequent pulses
- a SGL giving the duration of the pulse
- a SGL giving the delay of the pulse
- a SGL giving the duration of the recording (trace length)
- a SGL giving the sample rate
- a SGL giving the cycle time between iterations
- as often as iterations were performed:
  - a 1-D array of SGLs giving the voltage trace
  - a 1-D array of SGLs giving the current trace
- as often as iterations were performed:
  - a SGL giving the current value used for the IV-plot
  - a SGL giving the voltage value used for the IV-plot

8.2.2. Gsyn Threshold - *.GTH

- a SGL giving the length of the following string
- a string giving the template file used for the experiment
- a SGL giving the number of iterations done to determine thresh-g_{syn}
- a SGL giving the initial upper limit (Gmax) for the search algorithm
- a SGL giving the initial lower limit (Gmin) for the search algorithm
- a SGL giving the cycle time (ms) between iterations
- a SGL giving the reversal potential E_{rev} of the virtual synapse
- a SGL giving the sample rate (Hz)
- as often as the search algorithm executed:
  - a SGL giving the factor G used to scale the template
  - a 1-D array of SGLs giving the voltage trace

- a 1-D array of SGLs giving the current trace

8.2.3. Synaptic Gain - *.GAI; *.GAJ

GAI:

- a SGL giving the length of the following string
- string listing the template file(s) used for the experiment
- a SGL giving 1/actual sample rate (actual sample rate is in Hz)
- a SGL giving Erev of the virtual synapse
- a SGL giving the scale factor used for GT2[20]
- a SGL giving the length of the traces as # of points
- a number of bytes (0s) to extend header length to a total of 512 bytes
- voltage trace (SGLs)
- current trace (SGLs)

GAJ:

- a SGL giving the length of the following string
- string listing the template file(s) used for the experiment
- a SGL giving the length of the following string
- string listing the GAI-file corresponding to the experiment
- a SGL giving the reversal potential of the template file conductance(s)[#]
- a SGL giving the scale factor used for GT2[#20]
- a SGL giving the mean frequency of presynaptic activity (fpre)
- a SGL giving the gain value as determined in the experiment

If the GAJ-file contains data from a series of experiments, the information given above is repeated for each experiment.

[#] If the GAJ-file contains data from a series of experiments, these values reflect the settings for the last experiment in the series, i.e. if during the series reversal potential and scale factor are changed, these values will be incorrect for the experiments done so far.

[20] The scale factor used for GT2 is not saved at all!

8.2.4. V-clamp - *.PKP

Traces acquired with the V-clamp module are converted to the LabVIEW-specific data type 'Waveform' and saved as such. The 'Waveform' data type is a cluster of the following elements: a 1-D array of double precision floating point numbers (DBL – 8 bytes) making up a trace, a DBL giving the sampling interval and a DBL giving the time when the trace was acquired. In G-clamp the waveforms representing a voltage and a corresponding current trace are assembled into a 1-D array of waveforms and written to the file as one record. Traces from subsequent iterations are appended to the file as new records.

8.3. Save locations

IVB-, GTH- GAJ- and PKP-files are saved in the user-specified data directory on the host computer.

GAI-files are saved on the embedded computer in a subdirectory of c:\ni-rt\g-clamp\data. The name of the subdirectory is the date of the experiment in the form *ddmmyyyy*, with *dd* is day, *mm* is month and *yyyy* is year.

9. Adding new modules to G-clamp

This section provides easy, step-by-step instructions for adding a new conductance module to G-clamp. While this does not require expert knowledge of LabVIEW programming, a basic understanding of LabVIEW is helpful. Instructions for adding a new experiment module to G-clamp are presented in the G-clamp programmer's guide.

9.1. Adding a new conductance module

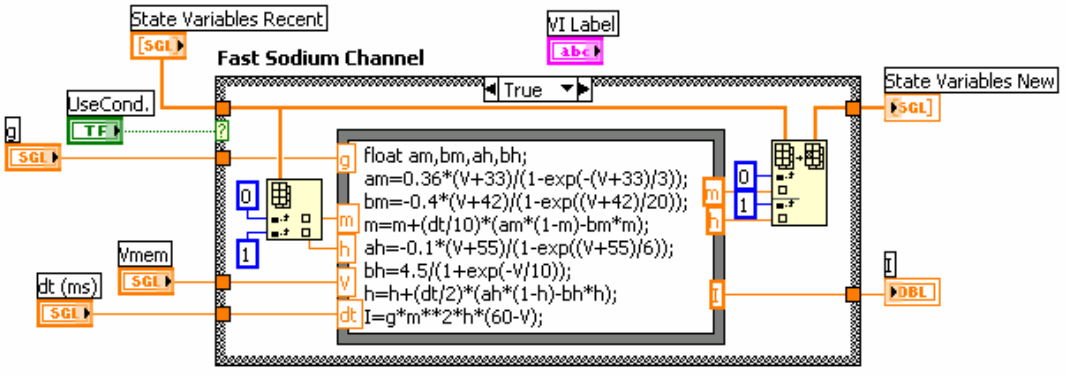
In the context of this paragraph, conductance module refers to an ionic conductance for which the conductance value g at any point in time is not known beforehand but has to be calculated in real time, e.g. a linear leak conductance or a Hodgekin-Huxley-type conductance.

The first step in creating a new conductance module is to save an existing module under a new name and to modify the function that describes the conductance.

1. On drive c: of the host computer create a directory ni-rt\g-clamp\gmodules.
2. From the directory c:\ni-rt\g-clamp on the embedded controller copy the file 'Embedded.llb' to c:\ni-rt\g-clamp on the host computer.
3. From the directory c:\ni-rt\g-clamp\gmodules on the embedded controller copy a conductance module VI to c:\ni-rt\g-clamp\modules on the host computer.
4. Start LabVIEW on the host computer (If you are prompted to specify a target platform, select *LabVIEW for Windows*).
5. Open the conductance module VI that was copied into c:\ni-rt\g-clamp\modules on the host computer. (Version 1 of G-clamp comes with the modules 'gKA.vi', 'gKdr.vi', gKM.vi', 'gLeak.vi' and 'gNa.vi'. Any of these can be used.)
6. Use *File* > *Save as*... to save the opened VI back into c:\ni-rt\g-clamp\modules on the host computer, but under a new name. Choose a name for the VI that makes its function easily recognizable, e.g. by following the naming convention outlined in step 5.
7. The front panel of the VI contains a string-control labeled 'VI Label'. The string in this control is the name with which the conductance appears in the list of virtual G's on the user interface of G-clamp. Edit this string to reflect the new conductance. To

make the change of the string permanent, first use O*perate* > *Make Current Values Default* or right-click on the string-control and select *Data Operations* > *Make Current Values Default*. Second, use *File* > *Save* to save the VI with the new default value.

8. Go into the diagram of the VI. The True-case of the central case-structure contains a formula node which contains the mathematical description of the conductance used to calculate g and subsequently I for the conductance.



The formula node has 5 inputs:

- **g** is the maximum conductance value as entered on the user-interface
- **m** and **h** describe the state of the activation and inactivation gate of the conductance. Values are between 0 and 1 and the actual input value is the value calculated by the preceding iteration of the dynamic clamp loop.
- **V** is the current membrane potential and
- **dt** is the time interval in ms between successive iterations of the dynamic clamp loop.

The formula node has 3 outputs:

- **m** and **h** are the new states of the activation and inactivation gates. These values will be used as input in the next iteration of the dynamic clamp loop.
- **I** is the current through the conductance as determined by m, h, V and the reversal potential of the conductance.

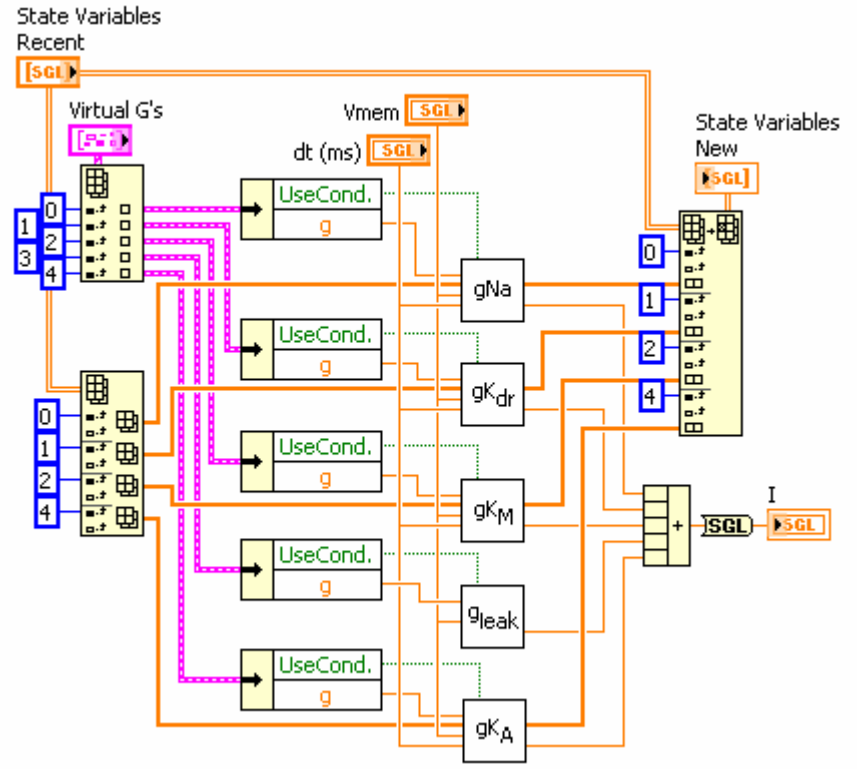
Modify the formulas within the node according to the mathematical description of the conductance you want to create. Keep in mind: The formula within the node executes like a script, i.e. it executes sequentially from top to bottom. Inputs do not need to be used, i.e. to create a non-inactivating conductance, simply do not use h in your

formulas[21]. Do not forget to set in the last line of the formula node the value reflecting the reversal potential.

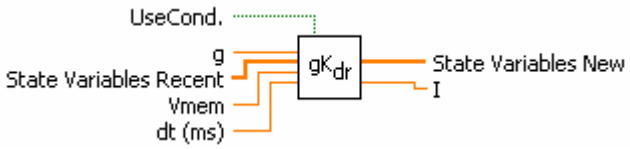
9. Save the VI.

The next step is to integrate the new conductance module into the real-time loop module. This is done by adding it to the 'V-dependent Conductances.vi', which bundles all conductance modules into one VI that is then used as a sub-VI in the real-time loop.

10. From the library 'Embedded.llb' open 'V-dependent Conductances.vi' and go into the wiring diagram.



Each of the conductance modules has the same pattern of input and output connectors[22]:



The inputs 'UseCond.' and 'g' are derived by unbundling a cluster which in turn is derived by indexing the 1-D array 'Virtual G's'. 'Virtual G's' essentially corresponds to the control 'Virtual G's' on the user interface. The input 'State Variables Recent'

[21] If h (and m) are not used, the input(s) can be deleted from the formula node (and the Index Array function shrunk accordingly). In that case, the corresponding output has to be deleted as well and the Replace Array Subset function shrunk accordingly.

[22] Because g_{leak} implements a linear leak conductance without activation and inactivation gates, the input 'State Variables Recent' and the output 'State Variables New' are not connected.

represents a 1-D array containing the values of m and h as calculated in the previous iteration of the real-time loop. This input is obtained by indexing a 2-D array of the same name, 'State Variables Recent'. 'Vmem' and 'dt (ms)' are directly provided by the real-time loop.

The output 'State Variables New' contains the updated values for m and h in a 1-D array. It simply replaces the old information contained in the 2-D array 'State Variables Recent', thereby creating the 2-D array 'State Variables New' that will provide the input to the 2-D array 'State Variables Recent' in the next iteration of the real-time loop. The output 'I' is summed up with the 'I' outputs from all other conductance modules, resulting in the summed output 'I'. In the real-time loop, 'I' is then added to any current due to simultaneous use of a conductance template or DC current step.

11. Add your new conductance module to the wiring diagram by using 'Select a VI…' from the LabVIEW functions palette and pick the newly created conductance module VI from c:\ni-rt\g-clamp\modules on the host computer.
12. Duplicate an 'Unbundle By Name' function and resize the 'Index Array' functions that operate on 'State Variables Recent' and 'Virtual G's' to add one input terminal. Provide the index input of these functions with a constant: the value of the constant should be [number of conductance modules – 1]. Similarly, resize the 'Replace Array Subset' function and the 'Compound Arithmetic' function. Provide the index input of the 'Replace Array Subset' function with a constant of value [number of conductance modules – 1]. Wire the inputs and outputs of the new conductance module to the resized functions following the same scheme as the existing conductance modules.
13. Save 'V-dependent Conductances.vi'.
14. Copy 'Embedded.llb' from the host computer back to c:\ni-rt\g-clamp on the embedded controller.
15. Copy the new conductance module VI from c:\ni-rt\g-clamp\modules on the host computer to c:\ni-rt\g-clamp\modules on the PXI controller.

16. Start G-clamp. G-clamp will detect the new conductance module, add it to the list of virtual G's on the user interface and update the configuration file containing the settings for the conductances. [23]

Important: Step 16 has to be performed before another conductance module is added to 'V-dependent Conductances.vi' and c:\ni-rt\g-clamp\modules on the PXI controller!

9.2. Adding a new experiment module

see G-clamp Programmer's Guide

10. Trouble Shooting

10.1. Jitter spikes of approximately 250 µs on the PXI controller while running LabVIEW RT

We encountered this problem with our PXI-8170 controller. The solution given below was found in the National Instruments KnowledgeBase.

Solution: You might have the USB hardware enabled on the PXI controller. To reduce the jitter in LabVIEW RT applications running on the PXI controller make the following BIOS setting changes[24]:

Integrated Peripherals >> USB Keyboard = DISABLED

For PXI-8170 controllers only:

PnP/PCI Configuration >> Assign IRQ for USB = DISABLED

It turned out that the PXI-8170 specific setting was the culprit in our case,

[23] 'Embedded.llb' and the directory c:\ni-rt\g-clamp on the host computer can now be deleted.
[24] This requires that the PXI controller is connected to a monitor and a keyboard.

10.2. Poor Network Performance in LabVIEW RT When Connected to Another Computer via a Crossover Cable

This was found in the National Instruments KnowledgeBase.

Problem: My LabVIEW RT engine is connected to another machine via a crossover cable. When I try to use TCP/IP or other networking protocols to transfer data back and forth, I see very poor performance. For certain packet sizes, my application may time out. What is wrong?

Solution: In some cases, an RT chassis can have poor network performance when connected to another computer via a crossover cable.

- **What is causing the problem?**
  An incorrect negotiation of the duplex state (half/full duplex) is occurring between the controller's Intel ethernet hardware and the other connected machine's ethernet hardware.

- **Why isn't it negotiating correctly?**
  Negotiation of half/full duplex is not always a guaranteed thing. Many cards, switches, and hubs have been known to incorrectly negotiate connections under some circumstances. It is not likely that this will happen, but it is possible.

- **Why does the incorrect negotiation kill network performance?**
  If one ethernet card thinks it is running full-duplex, and another thinks it is running in half-duplex, problems can occur. If the half-duplex configured card receives a frame while transmitting a frame (which, if the other end is full-duplex this is entirely likely), then it may interpret that as a collision. When collisions occur frequently, network performance becomes very poor.

- **Why does everything work fine with a hub and not a crossover cable?**
  Because both ethernet cards are negotiating the connection with the hub, and not with each other. In this case, the hub is correctly negotiating with both ethernet cards even though those cards fail to auto-negotiate correctly with each other.

- **What do I need to do to fix this problem?**
  Configure one of the cards to be forced half-duplex, rather than full-duplex or "Auto-negotiate". Often times Windows will allow you to configure the speed and duplex of an ethernet card. This works for most cards, but in some cases, the Ethernet Card ignores the Windows settings for speed and duplex. This is especially evident with the 3c905c and the Intel ethernet on the controller. If you manually specify the 3c905c to be 10Mbit (and not auto-negotiate), it will NOT change the link to 10Mbit (which can be witnessed by looking at the 100Mbit LED on the 817x controller - note that the LED will indicate that it is still on 100Mbit!). The same applies to the duplex. In order to change the settings of the 3c905c, you must use a DOS utility supplied by 3Com.

- **How do I change the 3c905c duplex mode?**
  Attached to this document is a floppy image that boots the computer into DOS. Once in DOS, run "3c90xcfg.exe". Then select "Install", then "Configure (F4)". It will give you options to configure on your 3Com card. Next either tab until the item list (that contains the driver mode, duplex mode, media type, etc.) becomes highlighted, or you can hit Alt+N to jump to it.

  The 3Com utility won't let you disable full-duplex AND keep the media type as auto-select. Instead, you have to turn off auto-select on the media type, and then it will let you disable (or turn off auto-select) full-duplex mode. So depending on the media type of your network, you'll want to select either 100Mbit or 10Mbit. Once you do so, the utility may switch the duplex mode to disabled. If it doesn't, you will want to switch it manually. Now hit the tab key until it highlights the "Ok" button. Hit enter, and it should say that it is writing the new configuration to the hardware. Once that completes, you may exit the program and reboot.

  To create the floppy, insert a floppy disk into your A: drive, and run the attached exe. It will prompt you for a floppy, and will format and copy this particular utility to it.

10.3. Sometimes saving of very long traces on the PXI controller fails

This problem affects only the Gain module. It usually happens when execution of a very long template is preceded by execution of a long but shorter template. This is a LabVIEW-inherent problem. The only solution to the problem is to re-boot the PXI-controller (and subsequently re-start G-clamp) before the very long template is executed.

This is the response we got from an applications engineer after we contacted National Instruments:

"Once you boot the PXI-controller, the embedded version of LabVIEW starts. The real-time operating system and LabVIEW get loaded into memory. As LabVIEW runs, it uses memory for compiled code and data. As arrays are built, memory gets allocated for these arrays. All elements in the array need to be contiguous in memory, without any breaks. If the program is stopped and then run again with the same array sizes, the same locations in memory can be used. However, if the program runs again with larger arrays, embedded LabVIEW has to look to previously unused locations in memory to find contiguous free space, since the previously used locations are not large enough. Even though the program (VI) had been stopped, the embedded LabVIEW did not stop, so the memory used by the VI did not get "wiped clean". Thus the workaround to "wipe the memory" between VI runs is to re-boot the PXI controller."