

Implementation of a Fast 16-Bit Dynamic Clamp Using LabVIEW-RT

Paul H. M. Kullmann, Diek W. Wheeler, Joshua Beacom, and John P. Horn

Department of Neurobiology and Center for the Neural Basis of Cognition, University of Pittsburgh School of Medicine, Pittsburgh, Pennsylvania 15261

Submitted 10 June 2003; accepted in final form 22 September 2003

Kullmann, Paul H. M., Diek W. Wheeler, Joshua Beacom, and John P. Horn. Implementation of a fast 16-bit dynamic clamp using LabVIEW-RT. *J Neurophysiol* 91: 542–554, 2004. First published September 24, 2003; 10.1152/jn.00559.2003. The dynamic-clamp method provides a powerful electrophysiological tool for creating virtual ionic conductances in living cells and studying their influence on membrane potential. Here we describe G-clamp, a new way to implement a dynamic clamp using the real-time version of the LabVIEW programming environment together with a Windows host, an embedded microprocessor that runs a real-time operating system and a multifunction data-acquisition board. The software includes descriptions of a fast voltage-dependent sodium conductance, delayed rectifier, M-type and A-type potassium conductances, and a leak conductance. The system can also read synaptic conductance waveforms from preassembled data files. These virtual conductances can be reliably implemented at speeds ≤ 43 kHz while simultaneously saving two channels of data with 16-bit precision. G-clamp also includes utilities for measuring current-voltage relations, synaptic strength, and synaptic gain. Taking an approach built on a commercially available software/hardware platform has resulted in a system that is easy to assemble and upgrade. In addition, the graphical programming structure of LabVIEW should make it relatively easy for others to adapt G-clamp for new experimental applications.

INTRODUCTION

Dynamic clamp is one of three methods in cellular electrophysiology that employs dynamic feedback to control the output of an intracellular recording amplifier. In classical current clamp, a feedback amplifier is used to create a current source the output of which follows a command waveform independent of its resistive load. In voltage clamp, feedback controls the current output to clamp the cellular membrane potential to a command waveform. By analogy with these earlier methods, dynamic clamp uses feedback to inject current that mimics the existence of a functional ionic conductance (g) in the cell membrane (Robinson and Kawai 1993; Sharp et al. 1993a,b).

Although the value of the dynamic-clamp technique has become widely appreciated (Dorval et al. 2001), the lack of standardized tools for its implementation remains problematic. Early approaches employed hardware and software that are now obsolete. More recent solutions exist (Butera et al. 2001; Dorval et al. 2001; Jaeger and Bower 1999) but adapting them to new experimental applications requires expertise in the DOS or real-time Linux operating systems coupled with programming skills in assembly languages and C. Our goal was to simplify dynamic-clamp experiments by developing a high-performance system that would be straightforward to put to-

gether and use. As part of the strategy, the application software was written in a high-level programming environment, thus making it easier to share and easier for others to modify. The result is G-clamp, a dynamic clamp solution based on LabVIEW-RT programming tools and associated hardware manufactured by National Instruments. The system runs under a standard Windows user-interface, and its performance can be upgraded as faster processors and data acquisition boards become available.

METHODS

Central technical problem and its generic solution

Because all currents through ion channels generally obey Ohm's law [$I = g * (V_m - E_{rev})$], creating a virtual conductance has three requirements: a description of the conductance (g) as a function of voltage and/or time, measurement of the membrane potential (V_m), and knowledge of the reversal potential (E_{rev}) for current through the conductance. These values are then used to compute a signal that drives the current command input of a current-clamp amplifier. Because of the reciprocal interactions between injected current, changes in V_m and the gating of voltage-dependent conductances (Fig. 1A), the feedback loop in a dynamic clamp (Fig. 1B) must run faster than the gating kinetics of the conductances under study. In many cases, this can be achieved with loop speeds in the range of 2–20 kHz. In this way, a dynamic clamp functions as a conductance clamp that effectively introduces a population of virtual ion channels into a living cell.

The central technical problem in building a dynamic clamp lies in the implementation of a fast feedback loop that must continually measure V_m , compute g and I_m , and then generate an output, one data point at a time (Fig. 1B). Although dedicated analog circuits could perform such calculations in real time, this approach is relatively inflexible and it requires specialized capabilities for design and fabrication (Le Masson et al. 1999; Mahowald and Douglas 1991). Consequently, most dynamic clamps employ digital computing, which presents other challenges, beginning with the operating system. The Windows, Unix, Linux, and Macintosh operating systems all provide convenient user interfaces, but they rely on interrupts that divide CPU time between multiple tasks, thus making them unsuitable for the precise timing required by the dynamic clamp's feedback loop (Fig. 1B). To resolve the timing issue, one group has developed a dynamic clamp that measures the duration of each cycle (Pinto et al. 2001). However, the variable cycle length resulting from this approach complicates the implementation of virtual channels having fast kinetics, and it precludes the use of predetermined conductance waveforms. A better solution is to use a real-time operating system that permits precise timing of the feedback loop with minimal cycle-to-cycle variation. Unfortunately, real-time operating systems can be difficult to install, lack many user-friendly features, and are sometimes not supported by hardware manufacturers. We overcame these

Address for reprint requests and other correspondence: P.H.M. Kullmann, Dept. of Neurobiology, E 1440 Biomedical Science Tower, University of Pittsburgh School of Medicine, Pittsburgh, PA 15261 (E-mail: pkullman@pitt.edu).

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked "advertisement" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

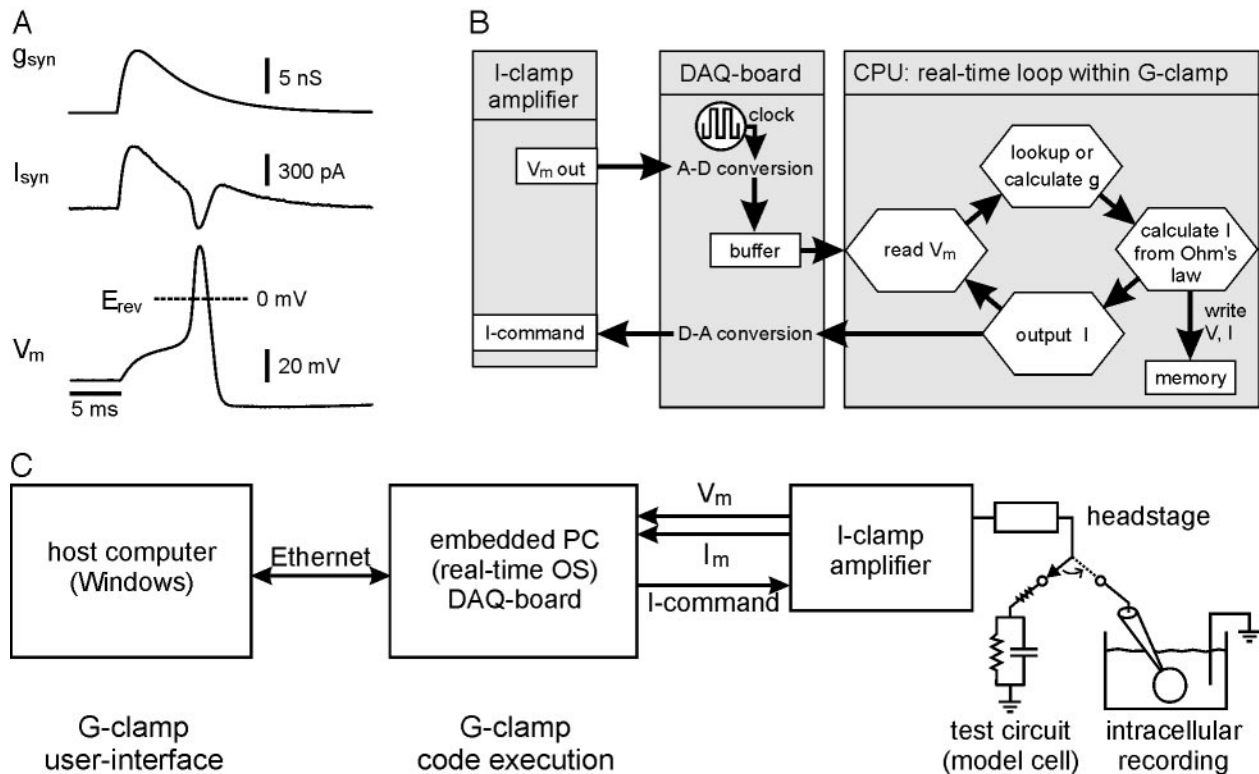


FIG. 1. An overview of dynamic-clamp methodology. *A*: reciprocal interactions between synaptic conductance (g_{syn}), synaptic current (I_{syn}), and membrane potential (V_m) illustrate the central technical challenge in building a dynamic clamp. In this example, I_{syn} does not simply follow the g_{syn} waveform because it also depends on the driving force, which changes with V_m . The disparity between g_{syn} and I_{syn} is most apparent when the excitatory postsynaptic potential (EPSP) triggers an action potential whose peak overshoots 0 mV. When V_m exceeds the equilibrium potential (E_{rev}) for the synaptic conductance (0 mV), it causes a transient reversal in the direction of I_{syn} . Current records in this and subsequent figures follow the standard current-clamp convention in which inward currents are depicted as upward and outward currents as downward. The data in *A* were from an experiment using a passive model cell together with a virtual nicotinic synapse and four other virtual conductances that give rise to a resting potential and the ability to generate an action potential. *B*: schematic organization of hardware interactions and the real-time feedback loop in G-clamp. Controlled by an onboard clock, the DAQ-board continually digitizes the V_m output of the recording amplifier at a user-specified rate and stores the values in an onboard buffer. The real-time loop starts by retrieving the oldest data point from the V_m buffer. The system then looks up or calculates the next value of virtual conductance and uses Ohm's law to calculate the next value of membrane current, which is written to memory and converted to an output command signal. The output drives the current-command circuitry in the recording amplifier, thereby completing one cycle of the feedback loop. *C*: organization of components in G-clamp. The system consists of a host computer, an embedded real-time computer and a current-clamp amplifier. The amplifier headstage can be connected to a passive test circuit or to a recording electrode.

limitations by using the LabVIEW real-time (RT) computing environment together with two computers (Fig. 1C). One computer running under Windows acts as the host platform for all programming and experimental control. The second computer runs a proprietary real-time operating system and performs as an embedded processor under control of the host. The embedded processor controls the multifunction board that conducts analog-to-digital (A-D) conversions for data recording and digital-to-analog (D-A) conversions for creating the current command signal. In addition to functionality, this arrangement provides convenience because all the programming, debugging, and button pushing is done on the Windows host. Once the program is ready, the RT engine on the embedded processor can interpret the code and be commanded to do its job without ever requiring the user to deal directly with the real-time operating system.

Commercially obtained components in G-clamp

The commercial components of G-clamp (Table 1) include an ordinary PC running Windows, a set of LabVIEW RT hardware and software components (National Instruments, Austin, TX), and an Axoclamp 2B intracellular recording amplifier (Axon Instruments,

Foster City, CA). All of the National Instruments components cost approximately \$10,000, which is similar to the cost of the recording amplifier.

National Instruments manufactures the embedded CPU as a modular industrial PC that plugs into a four-slot chassis built on the PXI/CompactPCI standard. The controller module includes a Pentium III processor, RAM, and a hard disk. In our present system, the second slot in the chassis holds an Ethernet card for communicating with the host computer and the third slot holds the multifunction data-acquisition (DAQ) board with 16-bit A-D and D-A converters and programmable gain. The DAQ board is wired to the V_m and I_m outputs and to the I_m -command input of the recording amplifier through a coaxial cabling interface. Signal gains have been optimized for maximal sensitivity and high dynamic range (Table 2) by matching the amplifier specifications to I/O gain settings that are controlled through the G-clamp program. The embedded processor comes from the manufacturer with its proprietary RT operating system and the LabVIEW-RT run-time engine already installed. Final assembly of the hardware and installation of the LabVIEW Development System and the LabVIEW Real-Time Module on the host computer are straightforward and clearly documented by National Instruments.

TABLE 1. *Commercially obtained components of G-clamp*

Component	Part	Main Characteristics/Comments
Host computer	300 MHz Pentium II with 128 MB RAM, 8 GB hard drive and 10/100 Ethernet adapter Windows NT4 LabVIEW 6.1 Full Development System* LabVIEW Real-Time Module*	According to the manufacturer, LabVIEW 6.1 runs on all versions of Windows from 9x through XP. When building G-clamp, we simply used an existing machine with NT4. Additional information concerning specifications of the host can be found in LabVIEW RT documentation (www.ni.com) The Ethernet connection between host and embedded controller can run through a LAN or directly between the two machines.
Embedded controller	PXI-8170* Real-Time Embedded Controller PXI-8211* PXI-6052E* PXI-1002*	850 MHz Pentium III processor, 192 MB RAM, 6 GB hard disk with real-time operating system and LabVIEW-RT engine installed by manufacturer 10/100 Ethernet card to communicate with host-computer Multifunction I/O Card with 16 analog-in and 2 analog-out channels, 16-bit A/D and D/A conversion PXI/CompactPCI-chassis for the PXI modules
Coaxial cable interface	BNC-2090*	Wiring interface for connecting the Multifunction card on the embedded computer to the recording amplifier.
Recording amplifier	Axoclamp-2B** with HS-2A 0.1x gain headstage	Designed for current-clamp recordings with patch-clamp or sharp microelectrodes

* National Instruments, Austin TX; ** Axon Instruments, Union City, CA.

Functional architecture of G-clamp software

G-clamp has a hierarchical modular structure (Fig. 2) that facilitates sharing between users and is open to future development of new applications. The user interface (Fig. 3), residing at the top level on the host computer, serves to select and control the experiment and to display its results. On starting an experiment, the user-interface module calls the LabVIEW run-time engine on the embedded CPU and commands it to execute one of three experiment modules. Unlike the user-interface module, all of the other software modules (Fig. 2) reside on the hard disk of the embedded processor and are loaded into the embedded memory as needed. This arrangement enhances performance by circumventing the need for continual downloads over the Ethernet link from the host hard drive. Each of the experiment modules runs a specific experimental protocol the purpose of which reflects our particular research on synaptic integration, but the structure of which could be adapted to related applications.

The first experiment module sends out a series of current pulses and uses the results to construct a steady-state *I-V* relation and measure its slope over a selectable range (Fig. 3). Unlike conventional *I-V* programs, this experiment can be run in the presence of one or more virtual conductances. Controls on the user interface allow for selection of virtual conductance parameters, for adjustments of pulse duration, pulse rate and related variables, and for the results to be stored as a data file after completion of the trial.

The second experiment module implements a virtual excitatory synapse and uses a binary search paradigm to scale the magnitude of the synaptic conductance until the excitatory postsynaptic potential (EPSP) reaches threshold for stimulating an action potential. By scaling a conductance waveform relative to this threshold, one can effectively normalize synaptic strength and more easily compare the behavior of different neurons.

TABLE 2. *Signal calibrations for G-clamp*

Axoclamp 2B Amplifier		National Instruments I/O digitizer setting	Signal Calibration	
Parameter	Sensitivity		Range	Sensitivity
V_m output	10 mV/mV	±2.5 V (A-D channel 0)	±250 mV	76.3 μV
I_m output	100 mV/nA	±1.0 V (A-D channel 1)	±10 nA	0.31 pA
I_m command	1 pA/mV	±10 V (D-A channel 0)	±10 nA	0.31 pA

Values for I_m output and I_m command sensitivity reflect the HS-2A headstage with a gain of 0.1. The headstage gain and the digitizer settings can be adjusted from the user interface.

The third experiment module simulates specific patterns of synaptic activity by reading long data files that contain templates describing appropriate conductance waveforms. In this experiment, the strength of synapses in the template file can be scaled using threshold results from the second experiment. At the end of the experiment, the program automatically calculates synaptic gain by counting the number of action potentials elicited by a given pattern of virtual EPSPs and dividing their frequency by the presynaptic firing rate associated with the template file. 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.

Serving as the heart of the G-clamp feedback loop, the RT module interacts with the experiment modules, the I/O hardware, and the virtual-conductance modules. The RT module executes each experimental protocol by using a hardware clock on the DAQ board to control A-D conversions and thereby drive iterations of the virtual-clamp loop (Fig. 1B). The loop module also makes the required calls to virtual conductance modules, generates analog output to the current-clamp command circuitry, and writes results to memory.

The virtual conductance modules work by evaluating kinetic equations and by reading files that contain predefined conductance waveforms. Equations describing a voltage-dependent Na^+ conductance, a noninactivating delayed rectifier K^+ conductance, an M-type K^+ conductance, and a linear leak conductance are based on a Hodgkin-Huxley style model used previously to describe the amphibian sympathetic B-type neuron (Schobesberger et al. 2000). The only difference from the previous model is that activation of the Na^+ conductance was slowed by a factor of 5 so that it would run accurately at a clamp speed of 20 kHz. The excitatory synapse in G-clamp is modeled after the nicotinic synapse found on B neurons. This synapse mediates a fast EPSP whose underlying conductance waveform can be described as the sum of two exponentials; an activation time constant of 1 ms and a decay time constant of 5 ms (Schobesberger et al. 2000). To illustrate the system's flexibility, we include in the results the implementation of a new conductance module for an A-type potassium conductance.

During each cycle of the dynamic feedback loop, G-clamp predicts the next value of every voltage-dependent conductance by solving the ordinary differential equations that describe them. The numerical integration is based on Euler's method, which involves a simple linear extrapolation of the appropriate gating variables. All of the calculations for evaluating equations are performed in RT, using the arithmetic operations and functions available in LabVIEW. Synaptic con-

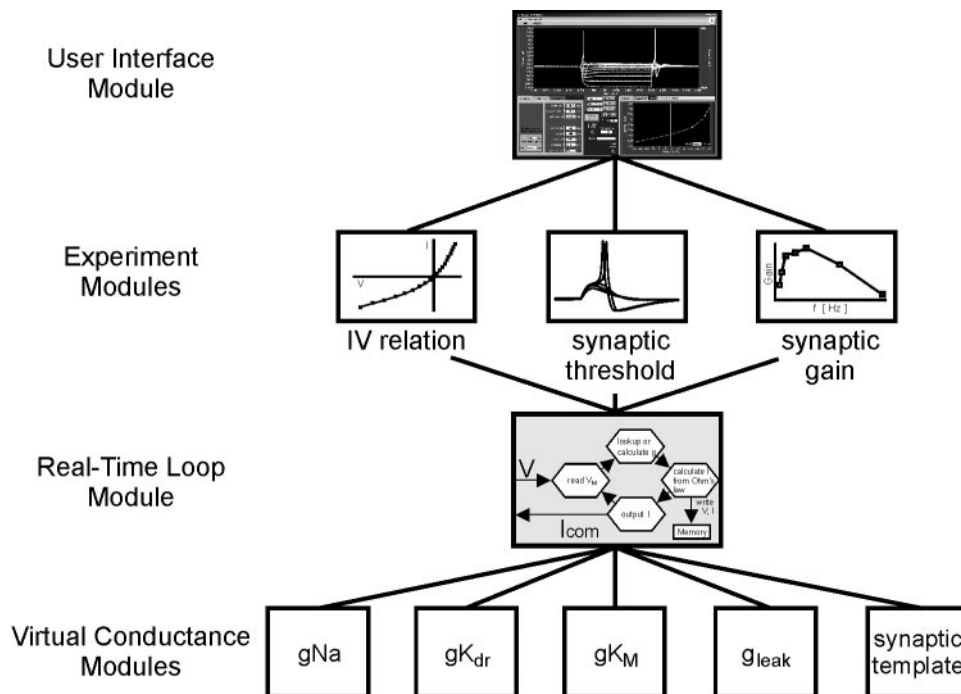


FIG. 2. Modular structure of the G-clamp program. At the top level, the user-interface residing on the host PC controls initiation of all experiments and displays results. All other modules reside on the embedded PC. One begins a session by selecting 1 of 3 experiment modules from the user-interface module and pushing the start button. All experiment modules use the same real-time loop module, which in turn draws on the virtual conductance modules. New experiment modules can be created by copying and modifying an existing module. The conductance modules have been written to accommodate additional types of conductances.

ductance is treated differently because we wanted to perform experiments that incorporated complicated patterns of synaptic activity. To accommodate this requirement, the system was designed to update the synaptic conductance during each cycle of the RT loop by reading the next value from a template file stored in memory. After determining an updated value for every conductance, the software then calculates the current that should flow through each virtual conductance, adds them, and sends out an appropriate signal to the constant-current command input. In this way, the system continually supplies the current needed to mimic one or more virtual conductances.

The synaptic conductance template files are prepared using NEUROSIM, a program that we have written in MATLAB (R12, The MathWorks, Natick, MA). Every file has a header that includes scaling information and parameters used to generate the template followed by a string of numbers that describe either a single synaptic event or complex patterns of synaptic events lasting several minutes.

The speed of the clamp is set in one of two ways. In experiments using conductance templates, the template controls the timing. Otherwise, clamp speed is set through the sampling rate control in the user interface (Fig. 3).

Distribution and installation of G-clamp software

G-clamp v1.1, as described in this paper, consists of two LabVIEW library files (LLBs), two dynamic link libraries (DLLs), and a read-me text file with additional instructions. This application software was developed with grant support from the National Institutes of Health, is freely available for research purposes, and can be obtained by contacting the authors. NEUROSIM, a MATLAB program for preparing synaptic template files and simulating synaptic activity in a conductance-based model neuron, is similarly available and will be described elsewhere in more detail. MATLAB simulations of action potentials used the ode15s equation solver.

In brief, installing G-clamp requires that one first create a folder named "G-clamp" anywhere on the host computer and a second folder named "G-clamp" in the NI-RT directory on the embedded PC. Copy the files "G-clamp Host.LLB" and "Ivanlys.DLL" into the G-clamp folder on the host computer. Use FTP to transfer the files "Embedded.LLB" into the folder "c:\NI-RT\G-clamp" and "nbifo.DLL" into the folder "c:\NI-RT\System" on the embedded PC. Enable the TCP/IP

protocol for the VI server of the LabVIEW-RT run-time engine and allow TCP/IP access by the host computer. Finally, set the instruction "UseSleepingTimedNonBuffered" to FALSE in the file "NI-RT.INI" on the embedded PC and the system is ready for operation.

Recordings from bullfrog sympathetic neurons

Paravertebral ganglia 8–10 were isolated from bullfrogs (*Rana catesbeiana*, 3–5 in), dissociated by enzymatic digestion with Liberase Blendzyme 1 (Roche Diagnostics, Indianapolis, IN), and the cells plated onto coverslips coated with poly-D-lysine (BD Biosciences, Bedford, MA). All procedures followed a protocol approved by the University's Institutional Animal Care and Use Committee. Cultures were maintained at room temperature for ≤ 1 wk in a diluted L-15 medium (65%) supplemented with 5 mM dextrose, 2 mM L-glutamine, 1 mM CaCl_2 , 5% fetal bovine serum, 100 U/ml penicillin and 100 $\mu\text{g/ml}$ streptomycin. Whole cell recordings were performed at room temperature with external solution containing (in mM) 115 NaCl, 2 KCl, 1.8 CaCl_2 , and 4 NaHEPES, adjusted to pH 7.3 with HCl. Patch pipettes had resistances of 2–5 $\text{M}\Omega$ when filled with internal solution consisting of (in mM) 110 KCl, 1 MgCl_2 , 0.4 CaCl_2 , 4.4 EGTA, 5 NaHEPES, 5 dextrose, and 2 Na_2ATP , adjusted to pH 7.2 with KOH.

RESULTS

Overview of dynamic-clamp performance

G-clamp was developed through an iterative process of design, programming, and evaluation. Here we begin by describing version 1.0 in terms of test results, comparisons with numerical simulations, and program demonstrations. This version of the program has five conductance modules (Fig. 2) the properties of which reflect data describing sympathetic neurons. The results from tests of this system document performance of the dynamic clamp, define its limits, and identify potential sources of experimental error that can arise from improper settings of the recording amplifier. We then describe a minor upgrade of the software (version 1.1) to illustrate how

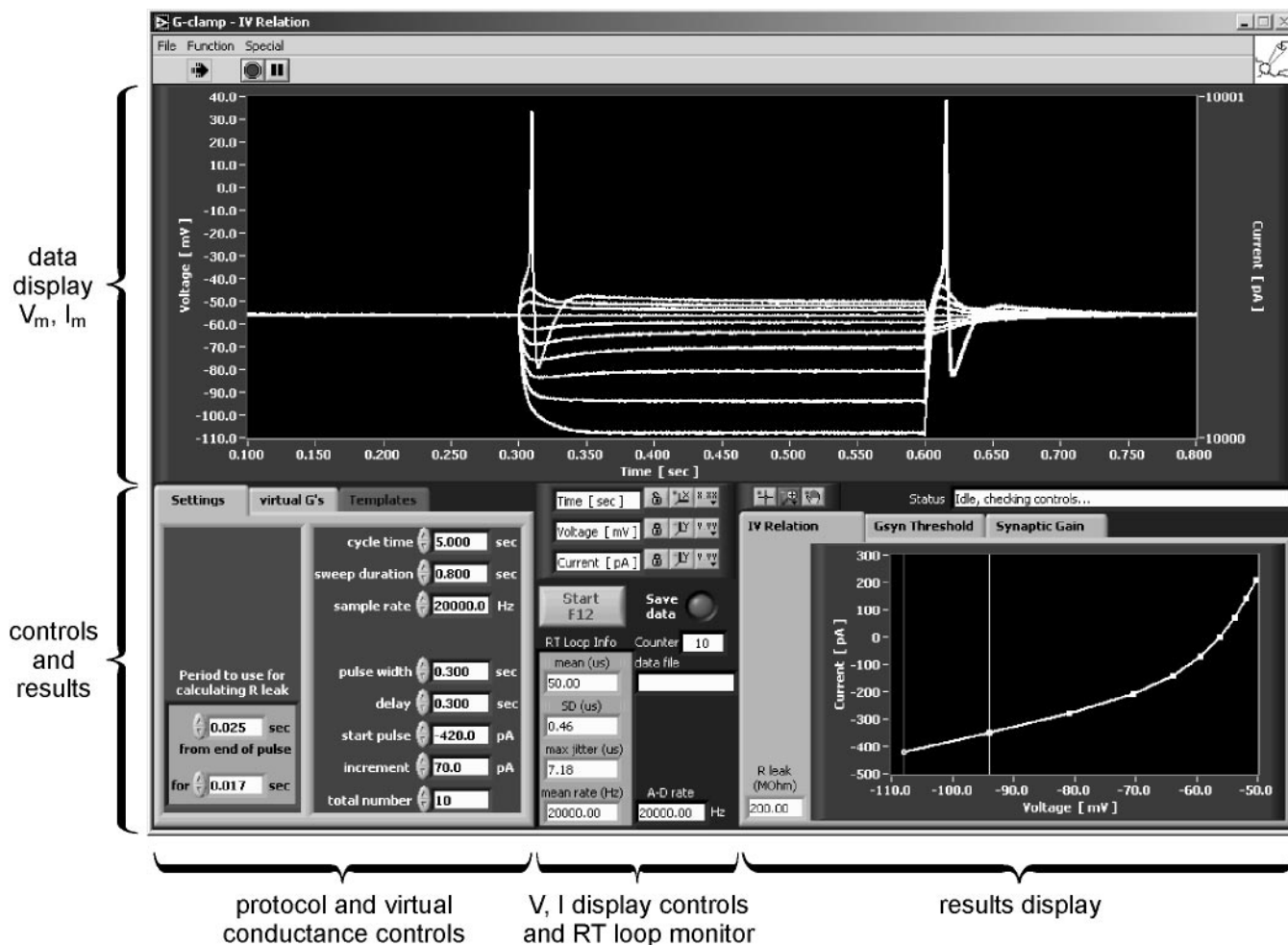


FIG. 3. Organization of the user-interface window. The window contains the elements needed to control a dynamic-clamp experiment and to assess its outcome. This example illustrates an *I-V* experiment, in which 4 virtual conductances have been added to a passive model cell. The top half of the window can display the recorded voltage and current signals. For simplicity, only V_m has been included in this example. The bottom half of the user-interface window includes 2 control areas and a results display. In the lower left corner, the 3 tabbed areas enable the user to control protocol settings and select virtual conductances. The lower middle area contains controls for the data display together with controls for starting and saving an experiment. This area also includes a display of real-time loop performance and digitizer performance that are used to verify that clamp timing is deterministic (see text). The lower right area of the interface can display results from each of the 3 experiment modules. In this case, it contains a steady-state *I-V* relation. The cursors define a segment of the *I-V* relation that is used to calculate the membrane resistance. In this example, the result defines the cell's leak resistance, which is displayed next to the graph.

the system can be extended to incorporate a new virtual conductance, which in this instance mimics the macroscopic behavior of A-type potassium channels.

Determinism of the RT loop

Each cycle through the RT loop in G-clamp, v1.0, (Fig. 1B) has a computational overhead that ranges from 2 to 113 arithmetic operations, depending on the type and number of virtual conductances being implemented (Table 3). Proper performance of the system requires that timing of this loop be reliable with minimal variation or jitter from cycle to cycle. Engineers describe these timing characteristics of a system as its determinism. To measure how fast G-clamp can run before determinism breaks down, a timer was incorporated into the RT loop. For this purpose we implemented the Read Time Stamp Counter (RDTSC) assembly instruction for Pentium-class processors, which can record from the processor the time of each

iteration through the loop. After the loop has finished executing, the time stamps are analyzed and the results displayed in terms of mean cycle length and jitter, defined as the SD and maximum deviation. In addition to loop performance data, the user-interface (Fig. 3) also displays the A-D sampling rate of the multifunction board. By comparing these outputs, we found

TABLE 3. Maximum real-time loop speeds achieved with G-clamp

Conductance	No. of Arithmetic Operations	Maximum Speed, kHz
Nicotinic synapse (g_{syn})	2	43.7
Linear leak conductance (g_{leak})	5	43.7
Muscarinic K^+ (g_{K_M})	24	40.2
Delayed rectifier K^+ ($g_{K_{dr}}$)	36	38.4
Fast Na^+ (g_{Na})	46	37.5
g_{Na} and $g_{K_{dr}}$	82	34.8
g_{Na} , $g_{K_{dr}}$, g_{K_M} , g_{leak} , and g_{syn}	113	32.7

that loop timing was deterministic when the measured loop rate agreed with the sampling rate. For example, when all virtual conductances were being used, the system could run at speeds ≤ 32.7 kHz before the RT loop failed to keep up with A-D conversions. Table 3 summarizes the maximum loop speeds attainable for different combinations of virtual conductances. Over the entire range of usable speeds, the SD of mean cycle time was $<0.5 \mu\text{s}$ and relatively constant, independent of the number of virtual conductances. These results identified the conditions where timing of the RT loop was reliable, but additional tests were needed to evaluate whether the dynamic clamp was correctly implementing virtual conductances.

Resting properties and excitability of an artificial neuron

Dynamic-clamp performance was evaluated by examining the resting properties and excitability of an artificial neuron with four virtual conductances (Fig. 4A). The combination of conductances and their magnitudes was chosen to resemble a model sympathetic neuron the behavior of which is well known from numerical simulations (Schobesberger et al. 2000). The system was tested by grounding the headstage of the current-clamp amplifier through an RC circuit (500 M Ω , 33 pF) in series with a 10 M Ω resistor (MCW-1U, Axon Instruments), a configuration that mimics a passive cell and recording electrode. After placing the amplifier into standard current-clamp mode, a current pulse was injected through the passive circuit so that the electrode resistance and stray capacitance could be nulled using the bridge balance and negative capacitance controls. The hybrid artificial neuron created by turning on the dynamic clamp consisted of the passive RC circuit in parallel with the virtual conductances (Fig. 4B).

The resting properties and excitability of the artificial neuron were almost identical to numerical solutions of the underlying equations obtained using a MATLAB program. In both cases, the resting potential was -56 mV. The artificial neuron (Fig. 4C) and MATLAB model (Fig. 4D) responded similarly to a series of current pulses. The steady-state I - V relations constructed from these two sets of data appeared identical when superimposed (not shown). One can also see from the records in Fig. 4, C and D, that the threshold and general shape of the action potentials were very similar when viewed on a relatively slow time scale. In addition to the agreement between the dynamic-clamp data and numerical simulations, two other observations supported the validity of the clamp's performance. First, the spike afterpotential reversed at E_K when the resting potential was varied systematically by injecting steady currents (not shown). Second, the virtual spike had a "physiological" positive overshoot, which peaked before attaining E_{Na} (Figs. 1A and 4C).

Although G-clamp action potentials and numerical simulations appeared similar, subtle differences between them became clear when the clamp's loop speed was varied and the results were viewed at sub-millisecond resolution (Fig. 5). Not surprisingly, the most challenging tasks for the dynamic clamp are to reproduce the upstroke and peak of the action potential because these events depend critically on the voltage-dependent Na^+ conductance the kinetics of which are the fastest process in the entire system. Comparing virtual action potentials recorded at different loop rates (2.5–30 kHz) revealed two important features of dynamic-clamp performance (Fig. 5). First, as loop rate was lowered, the upstroke of the action potential shifted leftward by $\leq 200 \mu\text{s}$ and its peak amplitude

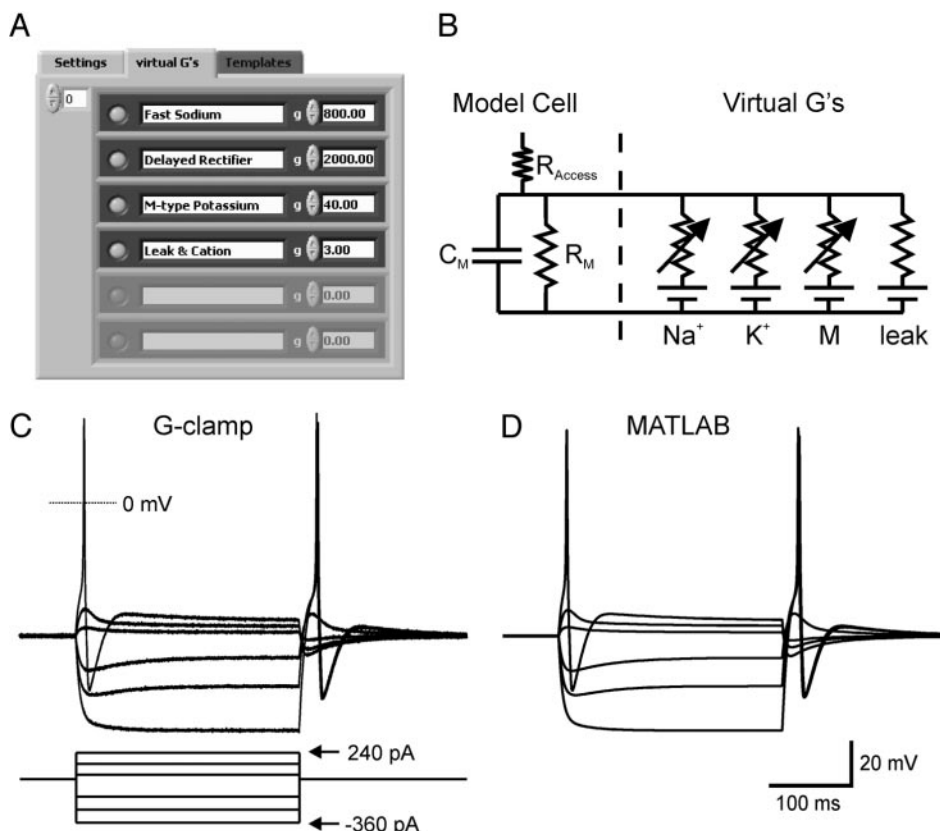


FIG. 4. Evaluation of steady-state I - V relation and excitability of an artificial neuron. *A*: 4 virtual conductances were turned on using the virtual G's panel in the user interface. In this dialog, the user also sets the peak magnitude of each conductance. *B*: when G-clamp is grounded through a passive model cell, an artificial neuron is created. *C*: response of the artificial neuron to 500-ms pulses of constant current. The system generates an overshooting action potential in response to depolarization and to the "anode break" at the end of a hyperpolarization. In this experiment, the clamp's loop rate was 20 kHz. *D*: simulation of the artificial neuron in MATLAB replicated the G-clamp data. The numerical simulation included the passive elements of the model cell and the 4 virtual conductances produced by the dynamic clamp.

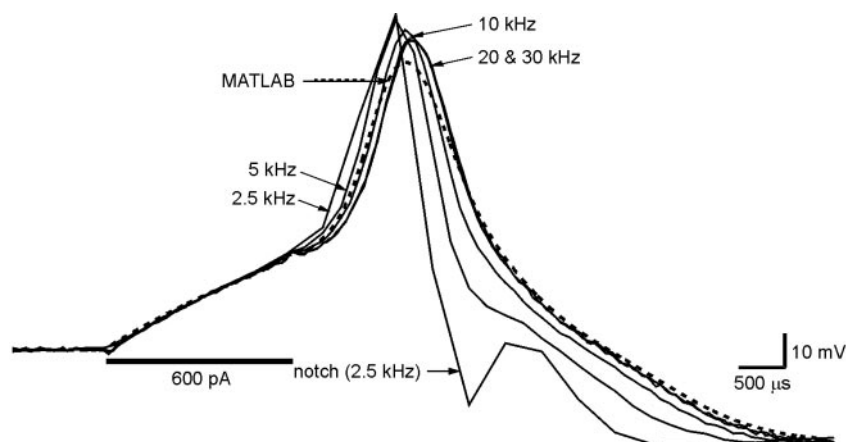


FIG. 5. Effect of the real-time loop rate on the virtual action potential. Action potentials triggered by 2-ms current pulses were recorded at 5 different loop speeds and compared with a numerical solution obtained using MATLAB. The virtual neuron was identical to that described in Fig. 4. The data show that increasing loop speed shifted the upstroke to the right and decreased the peak of the action potential. At the lowest loop speed, the action potential repolarization included a notch, which is an artifact. Note that the 20- and 30-kHz responses were almost identical. These 2 responses provided the best match with the MATLAB results, the main difference being a slightly lower peak in the numerical solution.

increased by ≤ 9 mV. Below 2.5 kHz, the system was unstable with the spike going positive to E_{Na} and the repolarization phase going into large oscillations. A less extreme version of this oscillation behavior appears as a notch in the 2.5-kHz response (Fig. 5). The effects of loop speed on spike shape arise in part from Euler's method, which G-clamp uses to integrate the conductance equations. The linear extrapolations employed by this method will give inaccurate results when the time steps between points are large relative to the rates of change in gating variables. Indeed similar problems will arise with other methods when the integration step size is too large. As a consequence of such errors, when the dynamic clamp runs too slowly, it overestimates g_{Na} near threshold and near the peak of the action potential, thus leading to a shift in the time of spike initiation and to an artifactual increase in spike amplitude. The second insight obtained from comparing loop speeds was that action potentials recorded at 20 and 30 kHz were nearly identical, thus demonstrating that G-clamp can produce a generic voltage-dependent g_{Na} at less than maximum speed. However, it is important to remember that the optimal loop speed for a given conductance will always depend on its particular kinetics.

We next examined how closely the G-clamp action potential matched a purely numerical solution at high temporal resolution. Figure 5 (---) shows that the MATLAB action potential peaks ~ 6 mV below the G-clamp spikes recorded at ≥ 20 kHz, but otherwise is very similar. To investigate this 6-mV difference, we conducted numerical simulations that were designed to mimic the Euler method, as implemented by G-clamp. In a simulation of the clamp running at a loop speed of 20 kHz, the output current was updated with a 30- μ s lag behind each voltage measurement, and the cellular potential was allowed to change continuously (i.e., 1 MHz) in response to 20-kHz current updates. The resulting action potential amplitude agreed well with the MATLAB result, differing by only 1.5 mV (not shown). The remaining 4.5-mV difference could not be explained by the choice of integration method used in the MATLAB computations. When the MATLAB solvers for stiff equations (ode15s, ode23t, ode23tb, ode23s) were compared, they all yielded similar results. The remaining 4.5-mV disparity between the G-clamp's action potential and purely numerical solutions of the equations must therefore originate elsewhere. The most likely source is inadequate electrode compensation, a view supported by experimental tests described in the following section.

Errors arising from improper compensation of electrode resistance and capacitance

Dynamic-clamp experiments often employ whole cell recordings with patch-clamp electrodes. The relatively low resistance and capacitance of such electrodes are generally easy to null out with a standard current-clamp amplifier. Nonetheless, it is quite likely that small errors in bridge balance or capacitance compensation will arise during real experiments. We therefore wanted to examine how such errors influence dynamic-clamp performance. In principle, one would expect bridge errors to be most problematic because they will lead to incorrect measurements of membrane potential across the entire frequency spectrum. By contrast, incorrect capacitance compensation should only influence high-frequency components of the voltage signal. Experimental tests confirmed this prediction. Overcompensating a 10-M Ω electrode by 4 M Ω can artifactually increase action potential amplitude and cause oscillations during repolarization (Fig. 6A). Overcompensating by as little as 1 M Ω is sufficient to increase action potential amplitude by 5 mV and account for the disparity between G-clamp recordings and MATLAB simulations. By contrast, undercompensating the bridge reduces the apparent size of the action potential but is stable and does not lead to ringing. As expected, improper capacitance compensation has smaller effects (Fig. 6B). Overcompensation generally causes the amplifier to ring, even without the dynamic clamp, and can be easily recognized and avoided. Capacitance undercompensation slows down the spike and increases its amplitude but produces little change in repolarization.

Behavior of the virtual nicotinic synapse

Having created a hybrid artificial neuron with a realistic resting potential and action potential, we next added a virtual nicotinic synapse whose reversal potential was 0 mV. As with the voltage-dependent conductances, the synaptic response was designed to mimic data from sympathetic neurons (Schobesberger et al. 2000). For these experiments, synaptic conductance waveforms were prepared in advance using a MATLAB program and stored as template files.

The conductance waveform required for a given experiment was selected from the templates menu (Fig. 7A). As would be expected, the time course of the conductance command signal differed from those of the synaptic current and EPSP that it

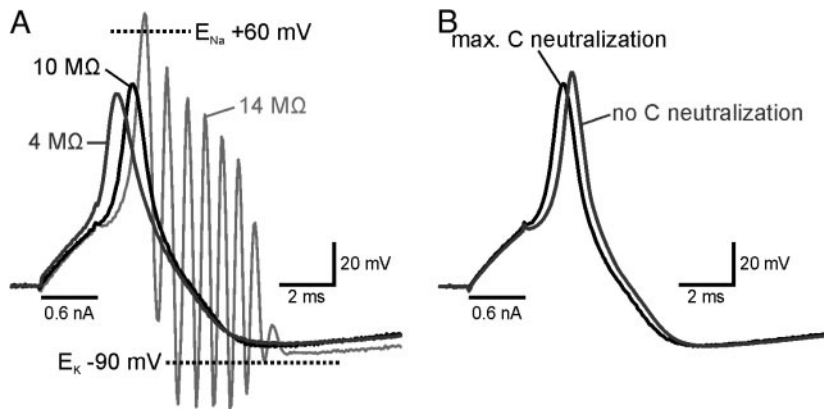


FIG. 6. Errors arising from improper compensation of electrode resistance (A) and capacitance (B). Tests were conducted using the artificial neuron described in Fig. 5 and a real-time loop speed of 20 kHz. A: when the electrode resistance (R_{Access}) of 10 M Ω was undercompensated (4 M Ω), the voltage response to current injection was exaggerated and the action potential occurs earlier and results in a lower spike amplitude. Overcompensation (14 M Ω) caused the action potential to exceed the Nernst equilibrium potentials for Na⁺ and K⁺ and led to oscillations during repolarization. B: capacitance neutralization alters the action potential by changing the time constant of the system. Undercompensation of the electrode capacitance slows down the voltage changes, leading to a delay in the action potential and a slightly higher peak amplitude.

produced (Fig. 7B). For purposes of illustration, the synaptic strength in this example was chosen to generate an EPSP with a subthreshold active component, which disappeared when the voltage-dependent g_{Na} was turned off. When the numerical solution describing the EPSP was computed in a MATLAB simulation, it superimposed exactly on the G-clamp response.

The EPSP undershoot seen in Fig. 7B arises from small changes in g_K produced by the transient depolarization. By varying the loop speed, it was determined that G-clamp accurately implemented the nicotinic conductance at rates ≥ 5 kHz (data not shown). This is consistent with the fact that the fastest component of g_{syn} , its rising phase, has a time constant of 1 ms.

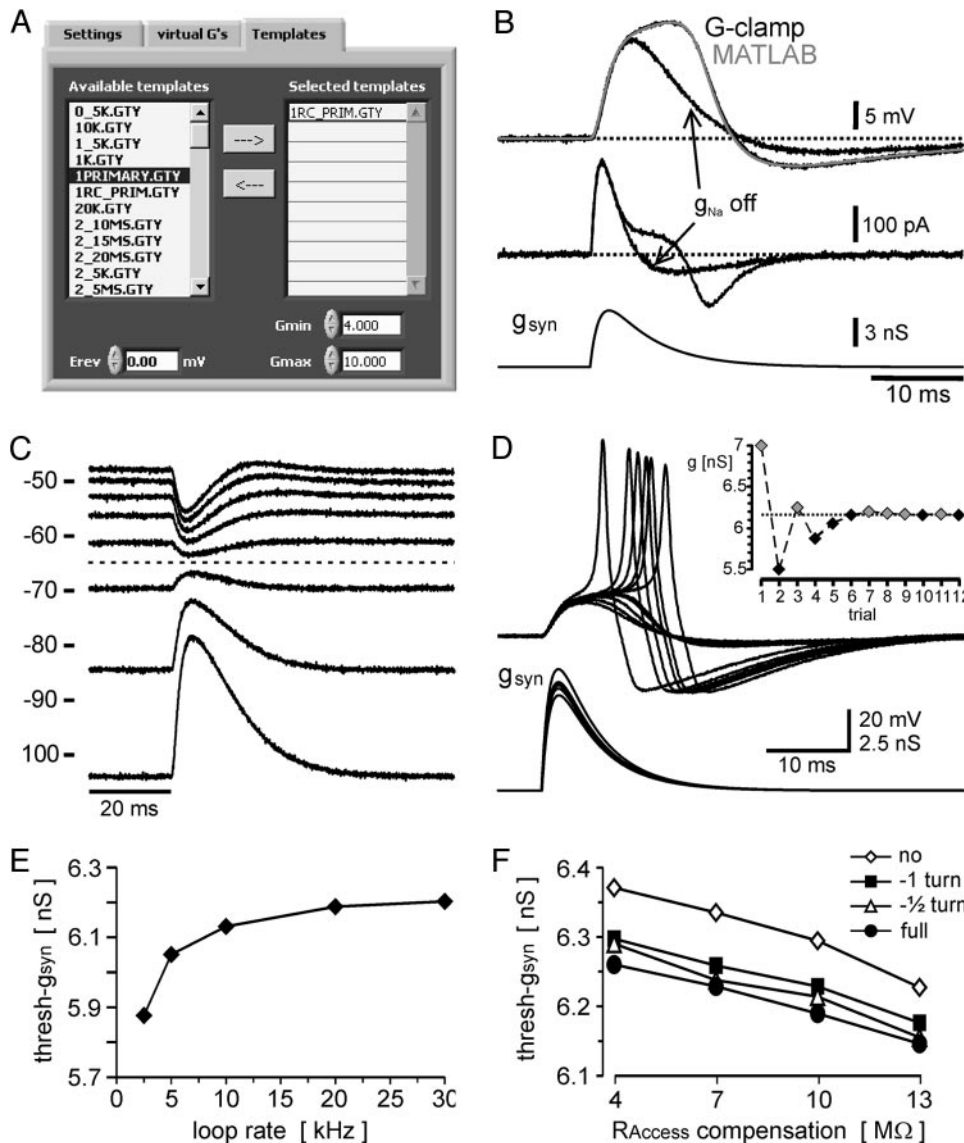


FIG. 7. Properties of virtual synapses. A: activating a virtual synapse in G-clamp is accomplished by selecting an appropriate conductance template file using the “Templates” dialog. This panel also includes settings for controlling the synaptic reversal potential (E_{rev}) and magnitude (G_{min} , G_{max}). B: behavior of a virtual excitatory synapse on the artificial neuron. The kinetics of g_{syn} , modeled after the nicotinic synapse on the bullfrog sympathetic B neuron, were stored in the template file. The peak synaptic conductance was set to 6.16 nS. With loop speed set at 20 kHz, the virtual synapse produced a subthreshold EPSP the active component of which was identified by turning off the voltage-dependent g_{Na} . Accuracy of the G-clamp result was checked by simulating the system in MATLAB. The numerical result (gray trace) was indistinguishable from the G-clamp voltage response. C: the excitatory nicotinic synapse was converted to an inhibitory synapse by changing E_{rev} to -65 mV. Systematic shifts of V_{rest} by steady current injection produced a family of inhibitory synaptic potentials (IPSP) that reversed at -65 mV (dashed line), as expected. D: the synaptic threshold module finds $thresh-g_{syn}$ by automatically rescaling an excitatory synaptic template depending on whether or not it elicits an action potential. For the search to succeed, the user must first enter values of G_{min} and G_{max} that straddle threshold. The algorithm generally settles on a solution within 10–12 trials (inset). E: measurement of $thresh-g_{syn}$ is relatively insensitive to the real-time loop rate when it is >5 kHz. F: improper compensation of electrode resistance and capacitance has only small effects on $thresh-g_{syn}$. Each line in the graph reflects a different level of capacitance neutralization. Every point in E and F is an average of 3 measurements.

To convert the nicotinic synapse into an inhibitory synapse, its reversal potential in the G-clamp program was changed from 0 to -65 mV, which is negative to spike threshold. When membrane potential was shifted by steady currents, the resulting family of inhibitory postsynaptic potentials (IPSPs) displayed appropriate voltage-sensitivity and reversal behavior (Fig. 7C). With these tools, one can therefore study virtual EPSPs, IPSPs and combinations of the two.

Measurement of synaptic strength

A central motivation for developing G-clamp was to study how virtual excitatory synapses interact on real neurons. This approach can be powerful because it permits synaptic events to be precisely defined in terms of their strength and timing. To define synaptic strength in a population of cells, it is useful to have a normalization method. For this reason, we have defined the threshold synaptic conductance ($\text{thresh-}g_{\text{syn}}$) as the conductance that is just barely strong enough to trigger an action potential (Schobesberger et al. 2000). G-clamp incorporates an algorithm for measuring $\text{thresh-}g_{\text{syn}}$. To use the routine, one must first estimate an upper boundary greater than the expected answer and enter it into the template window (Fig. 7A; G_{max}). The search range can be further limited by setting a lower

boundary >0 with the G_{min} control. The program then uses a binary search to find $\text{thresh-}g_{\text{syn}}$ in just a few trials (Fig. 7D).

To further evaluate the performance of G-clamp, we measured how estimates of $\text{thresh-}g_{\text{syn}}$ were influenced by loop rate and by electrode compensation. These tests revealed that over a fairly broad range of experimental conditions, the measurement of $\text{thresh-}g_{\text{syn}}$ was relatively insensitive to loop rate (Fig. 7E) and to errors in bridge balance and capacitance compensation (Fig. 7F).

Measurement of synaptic gain

Once $\text{thresh-}g_{\text{syn}}$ has been measured experimentally, this information can be used to scale the strength of virtual synapses in a more complicated experiment. For our studies of synaptic integration in sympathetic ganglia, we wish to model two classes of nicotinic synapse (Karila and Horn 2000). The strong inputs denoted as primary are expected to produce a postsynaptic conductance increase that is several times greater than $\text{thresh-}g_{\text{syn}}$. By contrast, the weak secondary synapses are associated with conductances that are less than $\text{thresh-}g_{\text{syn}}$. With G-clamp, one can take a template and easily scale the strength of synapses during an experiment. Figure 8 illustrates

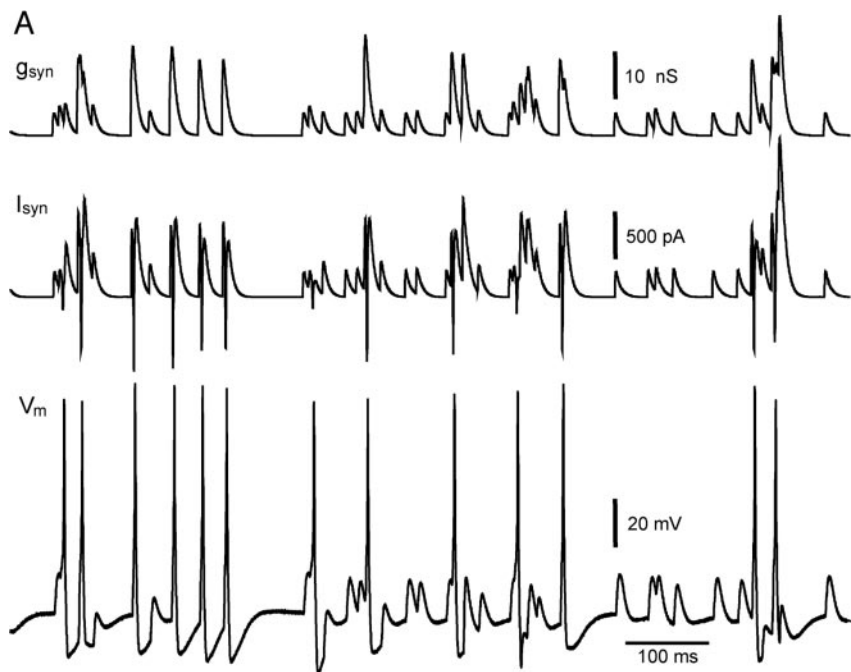
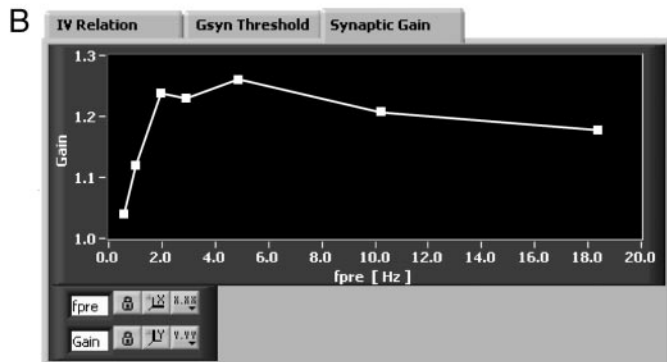


FIG. 8. Simulating virtual synaptic activity to measure synaptic gain. A: the g_{syn} waveform (top) reflects 4 synapses, each of which fires at an average rate of 10 Hz. It was created by independently scaling 2 templates and adding them together. The 1st template was scaled to mimic a single suprathreshold synapse and the 2nd to mimic the action of 3 subthreshold synapses. Middle: the total current flowing through the simulated synapses. The sharp downward deflections occur when action potentials overshoot and change the polarity of the driving force of I_{syn} . Inspecting the response of V_m to synaptic activity (bottom) reveals that some action potentials are driven by strong EPSPs from the primary synapse, whereas others are driven by summation of secondary EPSPs. The traces illustrate 1 s from a recording that lasted 20 s. B: G-clamp can automate the measurement of synaptic gain by sequentially stimulating a cell with synaptic templates having different frequencies of activity. The program records the data, detects and counts the postsynaptic action potentials, and then computes and displays the resulting gain relation.



an example in which the primary synapse was 15 nS and the three secondary synapses were each 5 nS. The noisy pattern of primary and secondary EPSPs was generated by assuming that activity arises from Poisson processes in which intervals between events at each synapse are exponentially distributed. The conductance template then elicited a series of synaptic currents that produced EPSPs, some of which triggered postsynaptic action potentials. By counting the number of presynaptic events and the number of postsynaptic action potentials, G-clamp calculates the synaptic gain of the neuron. The embedded processor's memory can accommodate synaptic templates lasting 330 s at 20 kHz while recording two channels of data. These long templates become important if one needs to study large numbers of synaptic events for statistical purposes in the design of experiments. With the synaptic gain module in G-clamp, one can automatically call up the templates, record and analyze synaptic gain over a range of presynaptic firing rates, and display the resulting synaptic gain relation (Fig. 8). In G-clamp, gain has been defined as the average postsynaptic firing rate divided by the average firing rate of each presynaptic neuron. The justification for this calculation derives from the synaptic connectivity of sympathetic ganglia (Karila and Horn 2000), but the analysis could be readily altered to reflect the organization of other neural circuits.

Recordings from bullfrog sympathetic neurons

After testing G-clamp by adding virtual conductances to a passive RC circuit, we next verified that the system would work on living cells by recording from dissociated sympathetic neurons with whole cell patch electrodes. In preliminary experiments using only the virtual nicotinic synapse, the system remained stable during physiological recordings and provided measures of threshold g_{syn} and synaptic gain that were consistent with theoretical predictions (Wheeler et

al. 2002). Figure 9 illustrates more challenging tests in which the voltage-dependent sodium conductance was added to or subtracted from two neurons. In a cell that was stimulated with a virtual suprathreshold EPSP, introducing increasing amounts of virtual g_{Na} systematically shifted the action potential to the left, produced a small increase in its peak amplitude and increased the maximum upstroke velocity without causing any instability in the system (Fig. 9A). These effects on the action potential's threshold and waveform are consistent with an increase in the total voltage-dependent inward current. In the converse experiment, subtracting virtual g_{Na} from an otherwise excitable neuron completely prevented the burst of action potentials generated by a prolonged depolarizing current pulse (Fig. 9B).

Modification of G-clamp to include a new virtual conductance

A central goal in developing this system was flexibility. Prompted by a request from colleagues, we recently implemented a transient A-type potassium conductance (g_{K_A}) whose kinetics were modeled using voltage-clamp data from mammalian midbrain dopaminergic neurons (Hahn et al. 2003). To create the new conductance, an existing conductance was cloned and its equations were modified to reflect the properties of g_{K_A} (Fig. 10, A and B). The new g_{K_A} module was integrated into the RT loop by linking it with the resized input and output functions, following the exact same scheme used with the existing conductance modules (Fig. 10A). Recordings from a sympathetic neuron illustrate how the virtual g_{K_A} can be used to manipulate a cell's repetitive firing behavior (Fig. 10C). Adding increasing amounts of virtual g_{K_A} enhanced the adaptation of firing, ultimately reducing it to a single spike. Taken together, the experiments with living neurons show that G-clamp can effectively implement virtual conductances, even

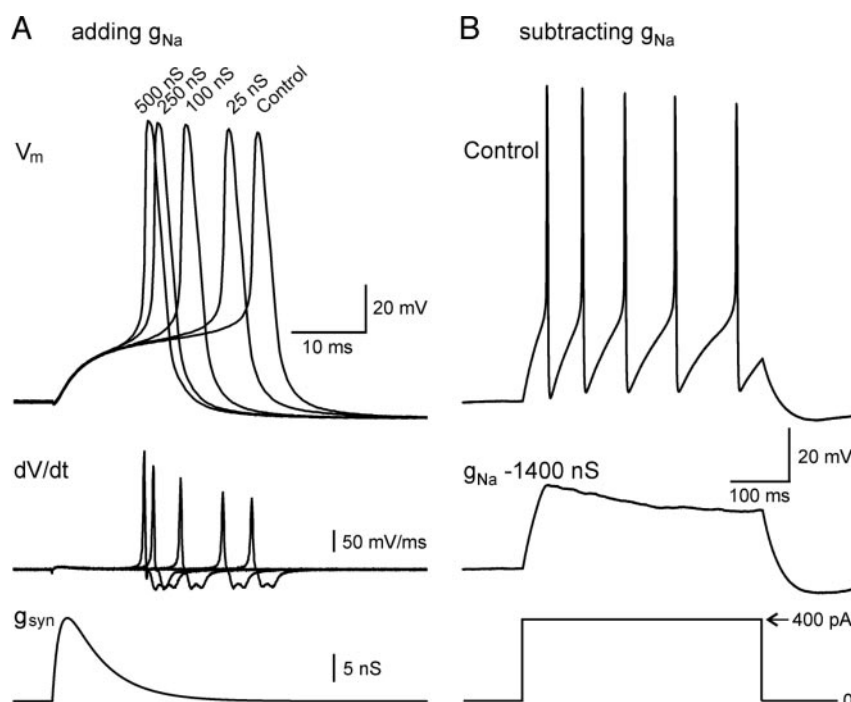


FIG. 9. Effects of the virtual sodium conductance, g_{Na} , on the excitability of sympathetic neurons. A: in 1 experiment, a virtual excitatory synapse (bottom, $g_{\text{syn}} = 15$ nS) elicited action potentials (top) in a dissociated sympathetic neuron. Adding g_{Na} over 5 trials progressively lowered threshold, increased action potential amplitude, and increased upstroke velocity (middle). $V_{\text{rest}} = -58$ mV. B: in another neuron, a constant current injection (bottom) elicited a train of action potentials under control conditions (top). Subtracting g_{Na} completely abolished action potential generation (middle). $V_{\text{rest}} = -69$ mV. Loop speed of the clamp was 20 kHz in both cells.

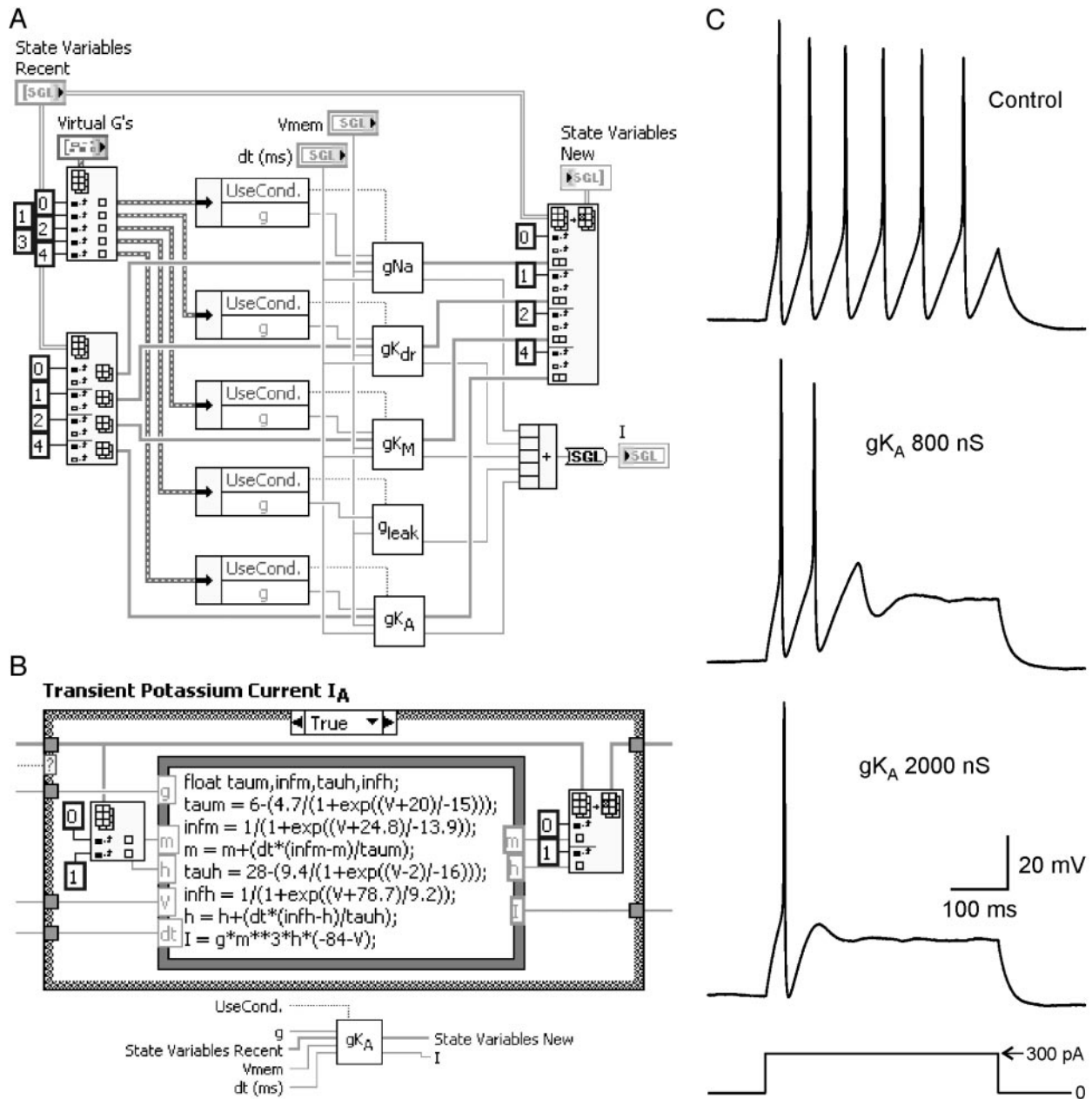


FIG. 10. Creation of a new virtual conductance in G-clamp. *A*: the portion of LabVIEW program code (“wiring diagram”) that calculates the current command signal from virtual conductances. A new conductance can be created by copying an existing virtual conductance module in this panel, modifying the formula describing the conductance (see *B*) and expanding the functions that provide the module with inputs and process its outputs. Using this approach, a new A-type potassium conductance, g_{KA} , was created in G-clamp. *B*: contents of the g_{KA} module. By clicking on the g_{KA} box in *A*, one can access the wiring diagram that contains the underlying equations describing the conductance and the forward Euler method for updating of gating variables. This script-like expression can be easily edited to change its properties or to generate a new virtual conductance. If the conductance is not in use, then the surrounding case structure executes the false option and outputs 0 for the current. The inset at *B*, bottom, illustrates how the g_{KA} module is connected with the rest of the program. *C*: recordings from a sympathetic neuron that illustrate the influence of g_{KA} on excitability. Under control conditions, the cell fired 6 action potentials (*top*) in response to a depolarizing current pulse (*bottom*). Turning on the virtual g_{KA} (*middle*) caused progressive accommodation in the repetitive firing behavior of this cell. $V_{rest} = -57$ mV; clamp loop speed = 20 kHz.

during the rapid voltage changes associated with action potentials.

DISCUSSION

This article documents the functionality of G-clamp, a dynamic-clamp system built using LabVIEW programming tools

together with an embedded PC that executes under a real-time operating system. The system was designed for high performance and ease of use. With G-clamp, it is possible to endow cells with artificial action potentials or simply to introduce fast virtual conductances. One can also use the system’s template files to study complex patterns of interaction between virtual

synapses. In addition to these capabilities, the architecture of G-clamp opens new avenues for future developments that will further enhance performance and allow for innovative experimental applications.

In developing G-clamp, the forward Euler method was chosen for numerical integration of differential equations because it is simple and therefore computationally efficient. Comparing the RT results obtained using a passive RC circuit with MATLAB simulations that employed more sophisticated integration methods revealed that the Euler method was both accurate and stable (Figs. 4, C and D, 5, and 6B). Further tests with sympathetic neurons confirmed that the system also performs well in biological experiments (Figs. 9 and 10C). However, it is conceivable that adding new virtual conductances will lead to problems. Because the forward Euler method can sometimes break down, it might become necessary to accept a higher computational load associated with a more stable integrator, such as the exponential Euler method (Bower and Beeman 1998). Such a change can be done easily by editing the equations in the conductance modules (Fig. 10B). One feature of G-clamp that may surprise new users is the perceptible delay between the execution of an experiment and the appearance of data in the user-interface window. This behavior is different from that of many commonly used data-acquisition systems that can rapidly display data as they are digitized and stored. The reason for this disparity lies in the nature of the RT embedded processor. During execution of the RT feedback loop, the system writes the data to memory, but does not transfer it to disk or to the host computer over the Ethernet link. For brief experiments such as the threshold search (Fig. 7D), this problem is minimal. However, the delay does become noticeable when measuring synaptic gain with long template files (Fig. 8). Thus it takes ~4 s to transfer the acquired traces of a 20-s recording to the host computer. It is therefore useful to monitor the progress of longer experiments with an oscilloscope or chart recorder. In future versions of the program, it should be possible to improve data transfers by including them in the RT loop when the system operates at lower computational loads. Data transfers will also improve as communication links become faster.

A powerful application of dynamic clamping involves the creation of virtual synapses between pairs of neurons or even small networks of cells (Bartos et al. 1999; Manor and Nadim 2001; Pinto et al. 2001; Sharp et al. 1996; Wilson et al. 2001). Performing such experiments with G-clamp will require additional channels for measurement of V_m and output of I_m command signals. Fourteen unused input channels and one additional output are available on the system's present I/O board. As the need arises, it will be possible to extend the G-clamp program to utilize these system resources. More output channels can also be added to the system by plugging another board with D-A channels into the unused slot in the backplane of the embedded computer.

Despite its virtues, dynamic clamping has been limited by its inability to mimic the electrotonic structure of neurons. More specifically, G-clamp incorporates the means for implementing multiple synapses but not for placing them at different sites on a dendrite. In the future, it may be possible to solve this

problem, at least partially, by designing virtual synapses on virtual dendrites (Dorval et al. 2001).

These and other ideas for further development of G-clamp will ultimately depend on the availability of computational speed and system memory. Fortunately, this is a generic problem that will benefit from general progress in hardware and software design. In the 2 yrs since this project began, a new generation of embedded processors with faster CPUs, and more memory has already become available. Although these recent enhancements in hardware performance are modest, the trend will likely continue. By transferring the current software and its successors to updated hardware, it should be possible to upgrade dynamic clamp performance with minimal effort. For these reasons, G-clamp may prove useful to other investigators and applicable to other dynamic-clamp experiments for at least several years to come.

ACKNOWLEDGMENTS

Data describing the A-type potassium conductance was provided by Dr. Ed Levitan.

Present address of J. Beacom: Keck School of Medicine, University of Southern California, Los Angeles, CA 90089 (E-mail: beacom@usc.edu).

GRANTS

This work was supported by National Institute of Neurological Disorders and Stroke Grant NS-21065 to J. P. Horn and by a postdoctoral fellowship from the American Heart Association, Pennsylvania/Delaware Affiliate, to D. W. Wheeler.

REFERENCES

- Bartos M, Manor Y, Nadim F, Marder E, and Nusbaum MP.** Coordination of fast and slow rhythmic neuronal circuits. *J Neurosci* 19: 6650–6660, 1999.
- Bower JM and Beeman D.** *The Book of GENESIS: Exploring Realistic Neural Models With the GENeral NEural Simulation System.* New York: Springer-Verlag, 1998.
- Butera RJ Jr, Wilson CG, Delnegro CA, and Smith JC.** A methodology for achieving high-speed rates for artificial conductance injection in electrically excitable biological cells. *IEEE Trans Biomed Eng* 48: 1460–1470, 2001.
- Dorval AD, Christini DJ, and White JA.** Real-Time linux dynamic clamp: a fast and flexible way to construct virtual ion channels in living cells. *Ann Biomed Eng* 29: 897–907, 2001.
- Hahn J, Tse T, and Levitan ES.** Long-term K^+ channel-mediated dampening of dopamine neuron excitability by the antipsychotic drug haloperidol. *J Neurosci* In press.
- Jaeger D and Bower JM.** Synaptic control of spiking in cerebellar Purkinje cells: dynamic current clamp based on model conductances. *J Neurosci* 19: 6090–6101, 1999.
- Karila P and Horn JP.** Secondary nicotinic synapses on sympathetic B neurons and their putative role in ganglionic amplification of activity. *J Neurosci* 20: 908–918, 2000.
- Le Masson S, Laflaquiere A, Bal T, and Le Masson G.** Analog circuits for modeling biological neural networks: design and applications. *IEEE Trans Biomed Eng* 46: 638–645, 1999.
- Mahowald M and Douglas R.** A silicon neuron. *Nature* 354: 515–518, 1991.
- Manor Y and Nadim F.** Synaptic depression mediates bistability in neuronal networks with recurrent inhibitory connectivity. *J Neurosci* 21: 9460–9470, 2001.
- Pinto RD, Elson RC, Szucs A, Rabinovich MI, Selverston AI, and Abarbanel HD.** Extended dynamic clamp: controlling up to four neurons using a single desktop computer and interface. *J Neurosci Methods* 108: 39–48, 2001.
- Robinson HPC and Kawai N.** Injection of digitally synthesized synaptic conductance transients to measure the integrative properties of neurons. *J Neurosci Methods* 49: 157–165, 1993.

- Schobesberger H, Wheeler DW, and Horn JP.** A model for pleiotropic muscarinic potentiation of fast synaptic transmission. *J Neurophysiol* 83: 1912–1923, 2000.
- Sharp AA, O’Neil MB, Abbott LF, and Marder E.** The dynamic clamp: artificial conductances in biological neurons. *Trends Neurosci* 16: 389–394, 1993a.
- Sharp AA, O’Neil MB, Abbott LF, and Marder E.** Dynamic clamp: computer-generated conductances in real neurons. *J Neurophysiol* 69: 992–995, 1993b.
- Sharp AA, Skinner FK, and Marder E.** Mechanisms of oscillation in dynamic clamp constructed two-cell half-center circuits. *J Neurophysiol* 76: 867–883, 1996.
- Wheeler DW, Kullmann PHM, and Horn JP.** Dynamic clamp analysis of synaptic gain arising from excitatory convergence in sympathetic ganglia. *Soc Neurosci Abstr* 146.8. 2002.
- Wilson CG, Butera RJ Jr, Del Negro CA, Rinzel J, and Smith JC.** Interfacing computer models with real neurons: respiratory “cyberneurons” created with the dynamic clamp. *Adv Exp Med Biol* 499: 119–124, 2001.