Command-line Electrophysiology
A Closed-loop Approach to Single Cell Characterisation

João Couto[1], Daniele Linaro[1] and Michele Giugliano[1,2,3]

[1]*Theoretical Neurobiology and Neuroengineering, Dept. Biomedical Sciences, Univ. of Antwerp, B-2610 Wilrijk, Belgium*

[2]*Dept. Computer Science, Univ. of Sheffield, S1 4DP Sheffield, U.K.*

[3]*Laboratory of Neural Microcircuitry, Brain Mind Institute, EPFL, CH-1015 Lausanne, Switzerland*

1 OBJECTIVES

Neuronal electrophysiology experiments are typically performed by using expensive commercial software packages and/or custom written applications: the latter are usually suited only for particular experiments and lack documentation. Additionally, the currently available packages, while focussing on usability by a large community of scientists, fail to provide proper scripting methods to systematically span the parameter space or to apply standard optimisation techniques while performing the experiment.

As neuroscience progresses, scientists are faced with questions that can only be answered by precisely triggering the stimulation to ongoing features of the experiment. This implies the usage of experimental paradigms that rely on short latencies (Prinz et al., 2004; Wallach et al., 2011) and whose implementation may require particular hardware and/or can be time consuming to develop from scratch. Recently, there has been growing interest in novel experimental paradigms based on real-time systems to perform closed-loop stimulation (Bettencourt et al., 2008; Lin et al., 2010; Zrenner et al., 2010): however, these approaches lack modularity and ease of integration with general scripting languages as they heavily rely on graphical user interfaces.

We developed a flexible software platform called LCG, whose main objective is to enable scientists to abstract from the technical details of the experiment while providing simple yet powerful tools to analyse online the recorded data and react accordingly.

LCG was developed with the following goals in mind:

- Serve as a general abstract platform for single cell and network electrophysiological experiments.
- Implement dynamic clamp (see (Robinson and Kawai, 1993; Sharp et al., 1993; Prinz et al., 2004) for a review) with active electrode compensation (Brette et al., 2008) to allow single electrode experiments.
- Allow relatively easy implementation of closed loop and hybrid experiments.
- Support multiple real-time engines in a standard desktop computer.
- Provide a simple installation procedure and support for scripting for on-the-fly parameter optimisation.

In the following sections we present the main features of LCG, which are relevant for performing *in vitro* or *in vivo* experiments employing both intracellular and extracellular recordings. As an application example, we describe a way to efficiently characterise the input-output relationship of single cells using a closed loop approach.

2 METHODS

The toolbox is written in C/C++ and consists of a library that can be linked to C/C++ programs written by the user. The software uses COMEDI to interface with the data acquisition card and supports two real-time Linux kernels, PREEMPT-RT and RTAI. It is worthwhile noting that, for performing standard voltage or current clamp experiments, the system does not require a real-time kernel.

The compilation of LCG generates two command-line applications that allow the user to perform non real-time (i.e., classical voltage and current clamp and extracellular recordings) and real-time experiments, respectively. This also implies that users do not need to have any knowledge of a programming language to effectively use LCG. The choice of a command-line interface (CLI) over a graphical user interface offers speed (an experienced user can perform the same task much faster using a CLI), stability (CLIs do not change between software updates) and scripting capabilities, enabling the user to automate a sequence

Couto J., Linaro D. and Giugliano ..
Command-line Electrophysiology - A Closed-loop Approach to Single Cell Characterisation.

of tasks by including them in a script. Additionally, the real-time program uses XML configuration files that *(i)* allow to programmatically explore physiological parameter ranges from a script, *(ii)* assure reproducibility since the configuration file can be stored with the recorded data and *(iii)* allow the development of complex protocols.

2.1 Toolbox Design Considerations

LCG is based on the concept of interacting *entities*: these are objects that can have multiple inputs, have one single output and can be connected to each other. Importantly, they implement the necessary logic to produce the output at each time step of the experiment. By making that every class in LCG is an entity and assuring that each entity performs only one specific task, code reusability is maximised, since complex experiments can be designed by combining multiple entities together. Entities can also communicate with each other asynchronously by emitting events: these are delivered to all connected entities and can be used, for instance, to communicate the occurrence of a spike or to send trigger and reset messages to other entities.
While it is possible to implement new custom entities in C++, LCG provides a set of entities that allow to perform common tasks such as data input and output, storing recorded data in compressed format, and generating arbitrary (current/voltage/conductance) waveforms. To perform hybrid experiments, LCG comes with a set of entities that implement commonly used neuronal and synaptic models.

3 RESULTS

We now discuss a simple application where LCG was used to record from pyramidal neurons in acute brain slices of the somatosensory cortex of juvenile rats.

One of the ways to characterise the input-output relation of single neurons is by computing the current-frequency relationship (f-I curve). This is traditionally done by applying several DC current steps of various amplitudes and measuring the corresponding firing frequency. As one has to let the cell recover between trials it can take several minutes to compute an f-I curve, and the curve is sampled coarsely since the current injected in the neuron is constant during each trial. We propose a novel approach to measuring the input-output relation by performing the experiment in closed loop using a proportional-integral-derivative (PID) controller. A sketch of the experimental setup is shown in Fig. 1A: a *Real Neuron* entity records the intracellular membrane potential of the cell under analysis and detects its spikes, which are sent, as events, to the *Frequency Estimator* entity. The output of the estimator is sent to the *PID controller*, which compares it to a target value, in this case a *Ramp* waveform. The controller, in turn, computes the current required to clamp the neuron to the target firing frequency and feeds it back to the neuron. The time course of the estimated firing frequency and of the injected current are shown in Fig. 1B, while the computed f-I curve is shown in Fig. 1C. The red curve is a first degree polynomial fit to the inverse of the ISIs (gray circles) and the green dots are the f-I curve computed with the “traditional” method. The computation of an f-I curve with the method presented here takes around 30 sec and provides a much better sampling than the traditional method. This is possible because the output of the PID is constant between spikes, thus allowing to associate each inter-spike interval (ISI) to a specific (constant) value of current, building the f-I curve shown here.

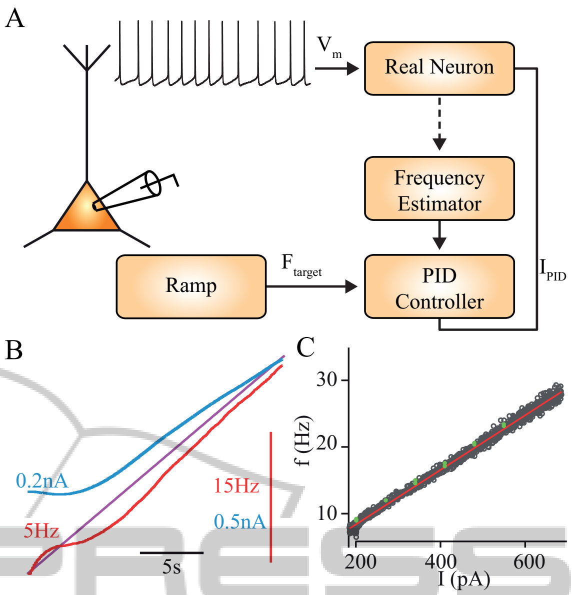

Figure 1: Example of computation of an f-I curve with a PID controller. **A** Schematic representation of the experimental setup (see text for explanation). **B** Example of the frequency estimate (red) and of the injected current (blue) to have the firing frequency follow a frequency ramp ranging from 5 to 30 Hz (magenta trace). The blue curve is the current injected by the controller. **C** Instantaneous firing frequency (inverse of the ISI) as a function of the injected current (gray circles). The red line is a linear fit to the data and the green dots are the f-I curve computed using the standard method.

This closed-loop approach is substantially different from the injection of a ramp of current to compute the f-I curve: in fact, in the latter case, the current injected between spikes is not constant, which makes associating an ISI with a particular current value difficult.

Additionally, this same experimental setup allows to clamp the firing frequency of a neuron at a given value, which can be useful in protocols that require delivering perturbations at fixed phases in the firing cycle of the cell (Miranda-Domínguez et al., 2010).

4 DISCUSSION

In this contribution we have presented LCG, a software to perform open- and closed-loop electrophysiological experiments. We have described some of the general principles that underly code reusability and flexibility of the toolbox. As a test case, we have shown a novel efficient way of measuring the input-output relationship of neurons by continuously estimating the firing frequency of the cell and, in closed-loop, clamping it to a desired frequency value. This allows to sample the f-I curve both with very high resolution and in a relatively short time (less than 30 s).

We are currently using LCG to perform both dynamic clamp experiments where very low latencies are required and *in vivo* experiments combining whole cell with extracellular recordings where large populations of neurons are probed simultaneously. Using LCG in conjunction with high-level scripting languages such as Python or Matlab offers several advantages, among which we mention:

- the possibility to use standard optimisation procedures to find, in real-time, optimal parameter values.
- The possibility to implement an experimental pipeline with standardised protocols, which allow to easily compare cells across experimental conditions and to speed up the successive data analysis.
- A relatively easy implementation of hybrid microcircuits, along the lines of what has been presented in (Kispersky et al., 2011).

It is worthwhile noting that, while some commercial and open source packages offer scripting capabilities, the possibility of performing electrophysiological recordings at the command line opens new possibilities for automating experimental workflows and allows users to closely integrate standard optimisation tools (for instance from Python's Numpy module) in their experiments.

By enabling closed loop experiments at various levels of latency and allowing to interface with general-purpose scripting languages, LCG has the potential to boost electrophysiological research to another level of automation and protocol complexity with minimal effort on the neuroscientist part.

ACKNOWLEDGEMENTS

D.L. is supported by the Flanders Research Foundation (grant no. 12C9112N, http://www.fwo.be). This work was partly supported by the University of Antwerp and by the European Commission, through the Seventh Framework Programme under the ICT - Future and Emerging Technologies scheme (project ENLIGHTENMENT, grant agreement no. 284801).

REFERENCES

Bettencourt, J., Lillis, K., Stupin, L., and White, J. (2008). Effects of imperfect dynamic clamp: computational and experimental results. *Journal of Neuroscience Methods*, 169:282–289.

Brette, R., Piwkowska, Z., Monier, C., Rudolph-Lilith, M., Fournier, J., Levy, M., Frégnac, Y., Bal, T., and Destexhe, A. (2008). High-resolution intracellular recordings using a real-time computational model of the electrode. *Neuron*, 59(3):379–391.

Kispersky, T. J., Economo, M. N., Randeria, P., and White, J. A. (2011). GenNet: A platform for hybrid network experiments. *Frontiers in Neuroinformatics*, 5:11.

Lin, R., Bettencourt, J., White, J., Christini, D., and Butera, R. (2010). Real-time experiment interface for biological control applications. In *Engineering in Medicine and Biology Society (EMBC)*, pages 4160–4163.

Miranda-Domínguez, O., Gonia, J., and Netoff, T. I. (2010). Firing rate control of a neuron using a linear proportional-integral controller. *Journal of Neural Engineering*, 7(6):066004.

Prinz, A. A., Abbott, L. F., and Marder, E. (2004). The dynamic clamp comes of age. *Trends in neurosciences*, 27(4):218–224.

Robinson, H. P. and Kawai, N. (1993). Injection of digitally synthesized synaptic conductance transients to measure the integrative properties of neurons. *Journal of Neuroscience Methods*, 49(3):157–165.

Sharp, A. A., O'Neil, M. B., Abbott, L. F., and Marder, E. (1993). The dynamic clamp: artificial conductances in biological neurons. *Trends in neurosciences*, 16(10):389–394.

Wallach, A., Eytan, D., Gal, A., Zrenner, C., and Marom, S. (2011). Neuronal response clamp. *Frontiers in neuroengineering*, 4:3.

Zrenner, C., Eytan, D., Wallach, A., Thier, P., and Marom, S. (2010). A generic framework for real-time multi-channel neuronal signal analysis, telemetry control, and sub-millisecond latency feedback generation. *Frontiers in neuroscience*, 4:173.