Xolotl: An Intuitive and Comprehensible Neuronal Simulator

Alec Hoyland [†]**, Srinivas Gorur-Shandilya** [†]**, and Eve Marder** [*]

Marder Lab, Brandeis University, Biology Department and Volen Center for Complex Systems, Waltham, MA, USA

Correspondence*:
Eve Marder
Volen Center for Complex Systems
Brandeis University
415 South Street
Waltham, MA 02454
USA
marder@brandeis.edu

[†] Co-first authors

2 ABSTRACT

3 `xolotl` is an open-source neuronal simulator written in `C++` with `MATLAB` wrappers. Complex
4 models and networks can be designed efficiently using an intuitive language tightly coupled to
5 the object-based architecture of the underlying `C++` code. Models can be specified by adding
6 conductances to compartment objects. The structure is modular, serialized, and searchable,
7 permitting high-level programmatic control over nearly all features of the models. `C++` templates
8 are provided for developing new conductances, compartments, and integration schemata.
9 It also includes a customizable graphical user interface ('puppeteer') for rapid prototyping
10 and hand-tuning conductances in real-time. The modular structure and accessibility to all
11 parameters, variables, and dynamics of the model network in `MATLAB` facilitate rapid construction
12 and assessment of model networks. `xolotl` is freely available at `https://github.com/`
13 `marderlab/xolotl`. This tool provides straightforward implementation and fast simulation of
14 neuronal models while permitting full control over every aspect of the network and integration.

15 **Keywords: simulator, MATLAB, C++, xolotl, conductance-based, computational, keyword, keyword**

1 INTRODUCTION

16 `xolotl` (`https://github.com/sg-s/xolotl`) is a fast single-compartment and multi-
17 compartment simulator in `C++` with `MATLAB` wrappers (`https://www.mathworks.com/`
18 `products/matlab.html`). Written with an emphasis on flexibility and speed, `xolotl` can simulate
19 single-compartment conductance-based models, networks of these, and detailed multi-compartment models.
20 `xolotl` exploits a novel automatic type system, `cpplab`, which binds `MATLAB` code to `C++` header
21 files, creating objects and classes *ad libitum* in `MATLAB` which reflect the underlying object-oriented code.
22 `xolotl` implements `cpplab` to represent the nested structure of conductance-based models, and exploits
23 the computational efficiency of the low-level programming language to quickly integrate models. For this
24 reason, models can be implemented entirely in `MATLAB` with few lines of code.

25 Models are specified in `MATLAB` by a `xolotl` object which contains compartment objects which
26 themselves contain conductances. Synapses belong to the `xolotl` object and connect compartments
27 together. The high-level specification supports arbitrarily large network and multi-compartment
28 morphologies.

29 `xolotl` provides parameter optimization capabilities through the algorithm-agnostic `procrustes`
30 toolbox. Any network parameters accessible through the `xolotl` structure can be optimized using arbitrary
31 algorithms and objective functions on multi-core computers and high-performance computing clusters.

32 The software has been implemented in `MATLAB` due to its ease-of-use and popularity among
33 neuroscientists. `cpplab` provides a powerful backend for specifying and integrating models without
34 relying on the significantly slower and limiting `MATLAB` `codegen`. Minimal experience with `MATLAB`
35 is required to use `xolotl`, and all equations and integration methods are provided transparently to the end
36 user. No string parsing of equations is required.

37 `xolotl` comes packaged with visualization functions, a graphical user interface for real-time
38 manipulation of model parameters. Plotting of voltage, intracellular calcium, conductance gating functions,
39 and time constants is provided by built-in `xolotl` methods. The `puppeteer` toolbox permits real-time
40 tuning of any network parameters using numerical sliders in a graphical interface which displays the
41 resultant membrane potential and intracellular calcium traces. The ease-of-use of these tools lends them to
42 pedagogical applications and rapid exploration of toy models. This tool aims to simplify the investigation
43 of dynamics of complex neural network models, facilitate collaborative modeling, and complement other
44 tools being developed in the neuroinformatics community.

2   ARTICLE TYPES

45 For requirements for a specific article type please refer to the Article Types on any Frontiers journal page.
46 Please also refer to Author Guidelines for further information on how to organize your manuscript in the
47 required sections or their equivalents for your field

3   WORKED EXAMPLES

48 Using `xolotl` in `MATLAB`, users create a `xolotl` object and populate it with compartments, synapses,
49 and controllers. Each field is a `cpplab` object constructed by a function call to `add`. The model
50 is integrated with the `integrate` function where the membrane potential, intracellular calcium
51 concentration, controller states, intrinsic currents, and synaptic currents can be outputs.

52 `xolotl` comes packaged with a library of pre-existing conductance and synapse objects which greatly
53 simplify the task of constructing model neurons. These objects can be referenced by name and added
54 directly to a compartment. Novel conductance dynamics can be easily written by modifying a template
55 header file contained in the `xolotl` distribution.

3.1   Simulating a Hodgkin-Huxley Model

57 The seminal Hodgkin-Huxley model contains a fast inactivating sodium conductance which promotes
58 spiking, a non-inactivating potassium delayed rectifer, and a passive leak current (1A). A compartment
59 named `HH` with compartment properties of membrane capacitance $C_m = 10\,\mu\mathrm{F/mm}^2$ and surface area
60 $A = 0.1\,\mathrm{mm}^2$ can be specified by 1B. Compartment, conductance, synapse, and controller properties can
61 be specified during the call to the `add` function, or after construction, using dot-notation in `MATLAB` (e.g.

62   `x.HH.Cm`). 1C shows the `MATLAB` command prompt after invoking the `xolotl` object `x`, displaying the
63   hierarchical structure inherent in conductance-based treatments of neurodynamics.

64   This model was constructed using conductances from Liu *et al* 1998. In the absence of applied positive
65   current, the model is quiescent and tonically spikes under 0.2 nA of applied current (1D). The `integrate`
66   function takes the applied current as an argument (e.g. `x.integrate(AppliedCurrent)`), so that the
67   `xolotl` object is agnostic to integration-specific perturbations. The `plot` function generates voltage and
68   intracellular calcium traces, where the voltage trace is colored by the dominant current. If the membrane
69   potential is increasing, the strongest instantaneous inward current colors the trace. Conversely, if the
70   membrane potential is decreasing, the strongest outward current colors the trace instead. 1F-I display the
71   results of the `show` function. Activation and inactivation steady-states and the voltage-dependent time
72   constants of these gating variables describe the conductance dynamics in absence of other channel types.

73 **3.2   Performing a Voltage Clamp Experiment *in-silico***

74   `xolotl` can recapitulate the results of voltage clamp experiments. 2 displays the procedure to clamp the
75   membrane potential of a cell with a delayed rectifier potassium conductance. The second argument of the
76   `integrate` function determines the clamped voltage (e.g. `x.integrate([], VoltageClamp)`).
77   Isolated currents under voltage clamp approach the steady-state (2D-E) so that a current-voltage relation
78   at steady-state can be extracted (2F). The derivative of the IV curve is the steady-state conductance (2G).
79   Fitting a sigmoid to various powers yields a model for the current dynamics (2H-I). These figures describe
80   graphically the theoretical underpinnings of current analysis through voltage clamp and can serve as an
81   effective pedagogical tool for computational and quantitative neuroscience.

82 **3.3   Simulating Network Models**

83   Network models in `xolotl` consist of compartment objects connected by synapses. Synapses are
84   stored in a vector array as a field of the `xolotl` object in `MATLAB`. Presynaptic and postsynaptic labels
85   indicate the connectivity of the synapse. Figure 3 implements a model of the triphasic pyloric rhythm in
86   the stomatogastric ganglion of crustaceans. The pyloric model contains three compartments and seven
87   synapses (3A). This structure is reciprocated in the hierarchy of the `xolotl` object, where conductances
88   are contained within compartments (3B). Synapses are upper-level properties of the network which point
89   between two compartments (3C). This exploits vectorized operations in `MATLAB` and does not require
90   each synapse to possess a unique name. The `plot` function generates multiple subplots when called for a
91   network with multiple compartments (3D-F).

92 **3.4   Simulating Integral Control**

93   `xolotl` can implement homeostatic tuning rules as integral control. Figure 4 depicts generation of a
94   bursting neuron model from quiescent conditions. Calcium sensors supervene on maximal conductance
95   density (4). In `xolotl`, integral controllers are properties of conductances (4B-C). They modify properties
96   of the conductance in response to an error signal. In a demonstration adapted from O'Leary *et al.* 2013,
97   integral control changes maximal conductances to bring a neuron from quiescence into a bursting regime.
98   Maximal conductances increase from random initial conditions to a set which elicits the desired network
99   output by minimizing the error signal (4D-F).

4 TECHNICAL DETAILS

4.1 Modeling

100

101 Models are specified by adding compartments and synapses to the `xolotl` object. Conductances are
102 added to compartments and controllers can be added to conductances. This modular structure recapitulates
103 the biophysics of the Hodgkin-Huxley formalism and obviates the need to explicitly write out equations,
104 which in `xolotl` are contained within the conductance header files.

105 `xolotl` relies on `cpplab` constructions, which allow the user to exploit the efficiency of low-level `C++`
106 code. `MATLAB` treats `cpplab` objects as standard variables allowing for symbolic manipulation using
107 only the high-level programming language and graphical interfaces. `xolotl` is fast specifically because
108 all time-intensive code is written in native `C++`. While automated `C++` transpiling from `MATLAB` using the
109 proprietary `coder` can drastically improve performance over loops through strong typing and memory
110 pre-allocation, supervenience of `MATLAB` over `C++` prevents efficient use of low-level features, such as
111 passing by reference and object-oriented programming.

112 Native `C++` provides drastic speed improvements beyond the benefits of translating `MATLAB` features
113 into low-level code. For this reason, `cpplab` has been designed to provide an interface for constructing,
114 transpiling, and compiling `C++` code to be called from within `MATLAB`. `xolotl` simulations can be
115 run entirely from `C++` executables. To facilitate use in `MATLAB`, `xolotl` uses the MD5 algorithm to
116 automatically hash the network and compile a new binary and `MEX` bridge file only if needed. `MATLAB`
117 provides a high level programmatic and graphical interface for implementing, manipulating, and visualizing
118 models without sacrificing the enhancements of native `C++` code.

4.1.1 Compartments and Synapses

119

120 Once the `xolotl` object is instantiated in `MATLAB`, the `add` function constructs new `cpplab` objects.
121 A model neuron consists of one or more compartments, each representing a section of membrane with
122 capacitance and surface area. Isopotential models require one compartment. Models with multiple neurons,
123 units, or non-trivial morphology require multiple compartments. All specifiable properties of compartments
124 are shown in Supplementary Table 1.

125 `xolotl` provides some features for generating complex models. Synapses can be added with the
126 `connect` function. At minimum possess identifiers to presynaptic and postsynaptic compartments and
127 default to electrical synapses. All specifiable properties of synapses are shown in Supplementary Table 2.
128 To create axons or transport chains, the `slice` function splits a compartment into n discrete segments and
129 adds these compartments to the network connected by electrical synapses.

CONFLICT OF INTEREST STATEMENT

130 The authors declare that the research was conducted in the absence of any commercial or financial
131 relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

132 AH and SG-S wrote the manuscript and the code. EM provided funding and moral support.

FUNDING

Details of all funding sources should be provided, including grant numbers if applicable. Please ensure to add all necessary funding information, as after publication this is no longer possible.

ACKNOWLEDGMENTS

The authors would like to thank Mara CP Rue and Hillary Rodgers for beta-testing the `xolotl` software.

SUPPLEMENTAL DATA

Table including all conductances packaged with `xolotl` should be put in the supplementary material.

DATA AVAILABILITY STATEMENT

The code to generate all figures can be found in the `xolotl` repository (`https://github.com/marderlab/xolotl`).

REFERENCES

[Dataset] LastName1, A., LastName2, A., and LastName3, A. (2011). Data title. doi:10.000/55555

LastName1, A., LastName2, A., and LastName3, A. (2013). Article title. *Frontiers in Neuroscience* 30, 10127–10134. doi:10.3389/fnins.2013.12345

Name, A. (1993). *The title of the work* (The city: The name of the publisher)

Name, C., Surname, D., and LastName, F. (1996). The title of the work. In *The title of the conference proceedings*, eds. E. Name1 and E. Name2 (The name of the publisher), 41–50

Surname, B. (2002). The title of the work. In *The title of the book*, ed. E. Name (The city: The name of the publisher). 201–213

Surname1, H. (2010). *The title of the work* (Patent country: Patent number)
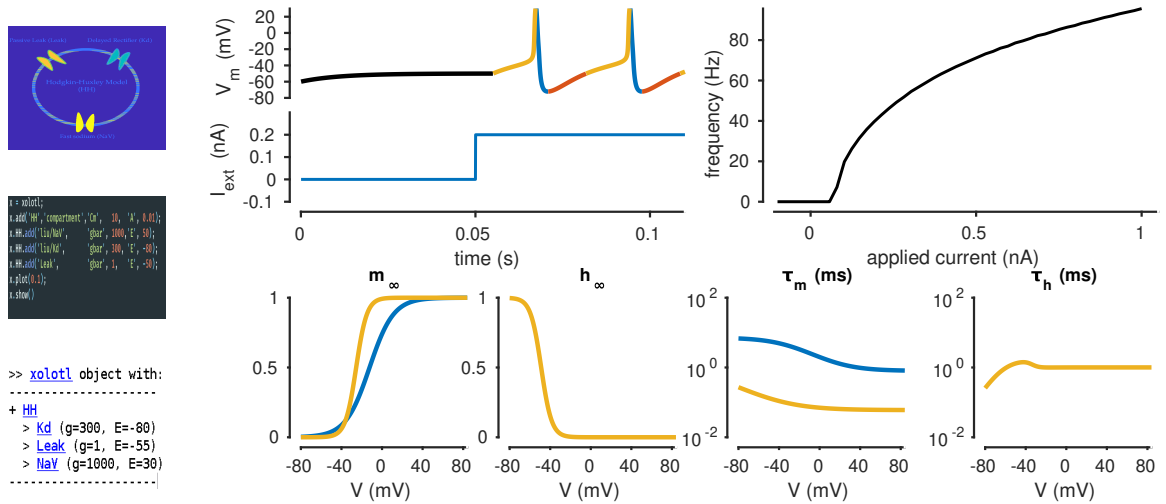
FIGURE CAPTIONS

Figure 1. `xolotl` can quickly set up and simulate conductance-based models. (A) Cartoon of a Hodgkin-Huxley single-compartment neuron model with fast sodium, delayed rectifier, and leak currents. (B) Code snippet in `MATLAB` used to implement D, F-I. (C) `xolotl` schematic displayed in the `MATLAB` command prompt. (D) Simulated voltage trace with 0.1 nA applied current. Colors indicate the dominant current (gold is fast sodium, blue is delayed rectifier, red is leak). (E) Frequency-input relation displaying firing rate as a function of applied current. (F-G) Steady-state gating functions for activation (m) and inactivation (h) gating variables. Variables not plotted are unity for all voltage. (H-I) Voltage-dependence of time constants for activation (m) and inactivation (h) gating variables. Variables not plotted are unity for all voltage. Colors indicate conductance type (gold is fast sodium, blue is delayed rectifier, red is leak).
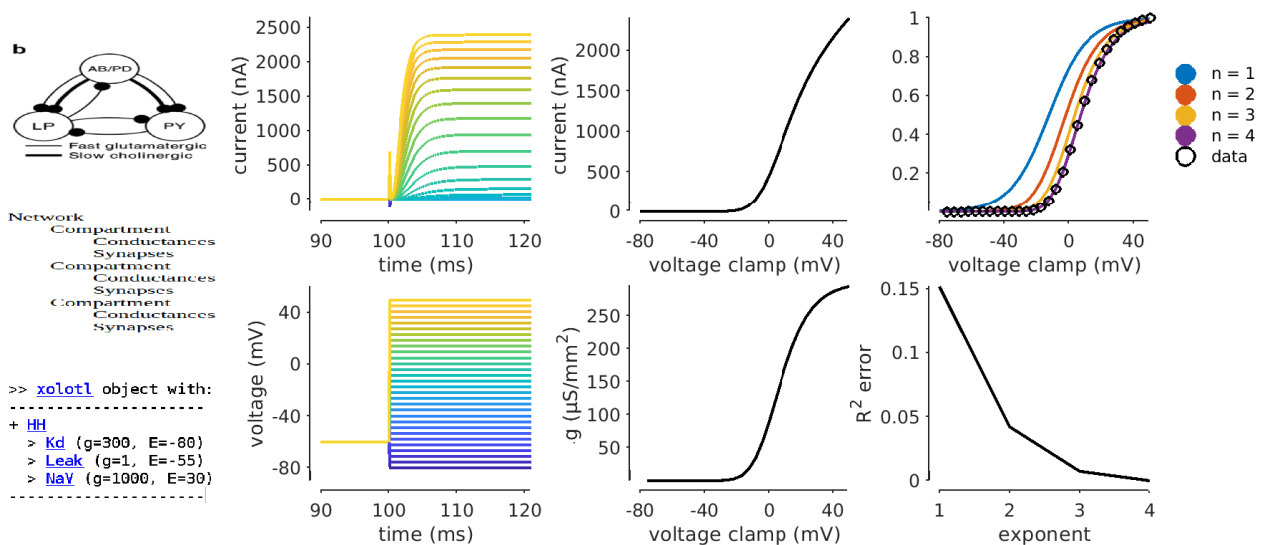

Figure 2. `xolotl` readily implements voltage clamp. (A) Cartoon of a cell with potassium conductance with experimentally-fixed voltage. (B) Structure of `xolotl` object in A. (C) Code snippet depicting integration under voltage clamp. (D-E) Current response to steps in voltage from a holding potential of $V_m = -60$ mV. (F) Current-voltage relation of the steady-state current ($t = 400$ ms) indicating a reversal potential of $E = -80$ mV and no inactivation. (G) Conductance-voltage relation at steady-state takes the form of a sigmoid. (H) Sigmoids m fit to the model as m^n data indicating that $n = 4$ is the best fit. (I) R^2 correlation of the sigmoid fits at various powers where $n = 4$ is an exact fit.
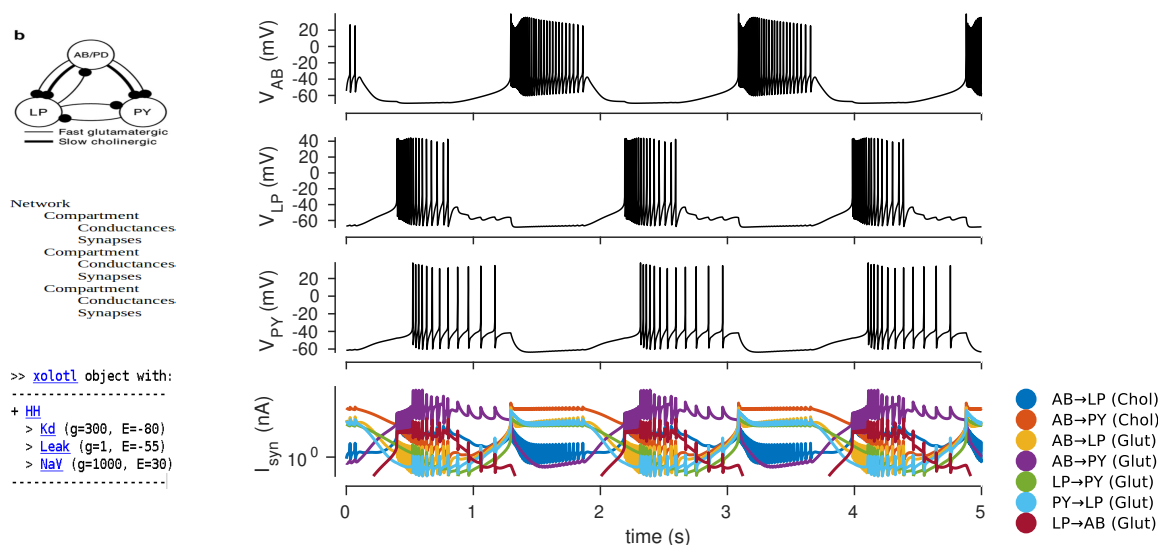
Figure 3. `xolotl` readily implements conductance-based network models. (A) Diagram of a network model of the pyloric rhythm in the crustacean stomatogastric ganglion (Prinz *et al.* 2004). (B) Hierarchical structure of a neuronal network considers compartments as components of the network and conductances and synapses as components of compartments. (C) *xolotl* implements conductances as fields of compartments and synapses as connections between compartments. (D-F) Simulated voltage trace of a model network for the three compartments. (G) Time series of synaptic currents in the simulated network with model cholinergic (chol) and glutamatergic (glut) synapses are outputs of the integration.
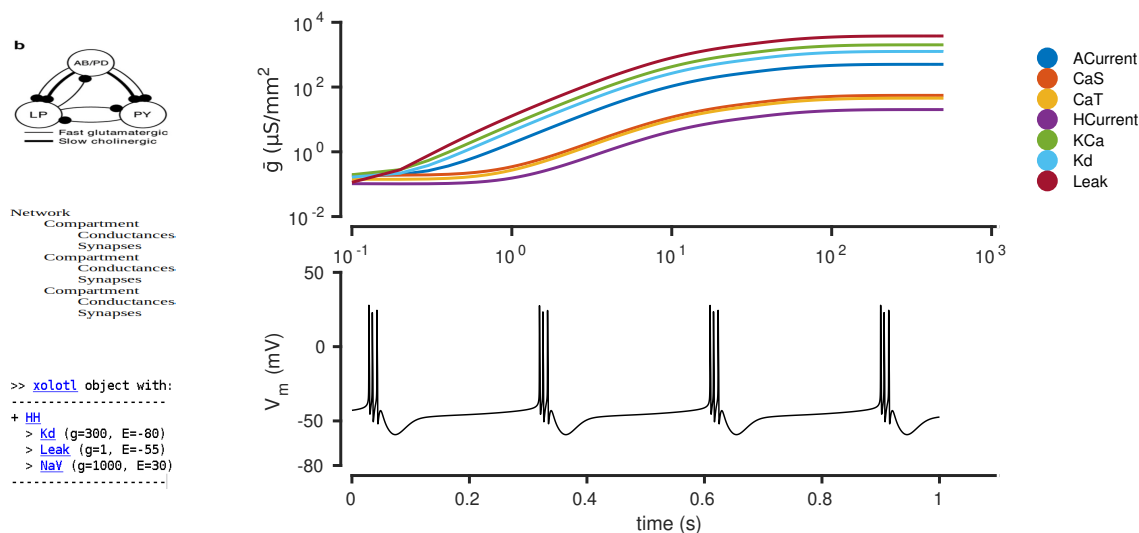

Figure 4. `xolotl` can implement homeostatic tuning rules as integral control. (A) Cartoon of a model neuron with integral control. (B) Hierarchical structure of a neuronal network considers controllers as components of compartments which act on conductances. (C) `xolotl` implements controllers XYZ. (D) Calcium sensors change maximal conductances to move a neuron from quiescence to a bursting state. (E) Voltage trace of the controlled neuron shows regular bursting activity.