Neurodemo V1.3.1

A Simulator for Learning and Understanding Electrophysiology Concepts

www.github.com/pbmanis/neurodemo

CMNB 2022A

Fall 2023

Document version 1.3

Table of Contents

Introduction

Neurodemo is a simulation program that lets you explore passive membranes, how action potentials are generated, and how trains of action potentials are generated. The program lets you control stimuli, as well as which ion channels are in the membrane, their density, and the reversal potentials for their ionic conductances. In addition, the program helps you to perform some simple analyses, export results to .csv files or figures, and plot internal channel states such as gating variables and open probabilities.
This program was originally written by Dr. Luke Campagnola, who was a graduate student in the Neurobiology Curriculum, and who helped to TA the electrophysiology block for several years. The program is written in Python. We have updated the code to work with the current versions of Python and support libraries.

We provide packaged installers that create Windows and macOS executables (apps), so you do not have to learn Python or how to work with the code directly. The advantage of using a program such as this is that it is cross-platform (Windows, macOS, Linux), does not depend on a continuous internet connection, and we can fix bugs as needed.

The simulation program will be briefly introduced in the first session of the class and covered in more detail in subsequent classes. We encourage you to become familiar with how to interact with it, both from the standpoint of controlling the simulations and interacting with the graphics. The design is possibly different than what you have used before, so it may take a few tries to become comfortable with how you interact with it. However, once you get the hang of the graphical interactions, you will find they give you a great deal of function with relatively little fuss.

In the remainder of this document, we describe how to interact with the program and then present several “experiments,” along with questions and manipulations you should work to answer and carry out. Some of this will be covered in class, but some will also be part of a problem set that will be graded.

Installation

Windows:

We have made a working executable file for Windows. All you should need to do is download the .exe file from Sakai to your desktop, and then run it. Windows will complain that this is unknown software, but you can sidestep this and "Run anyway". This should work on Windows 10 and 11. Updates will be supplied as new .exe files.

If you have difficulty or would prefer to run from the source code, you can follow the instructions in Appendix I.

macOS:

We have built two installers for macOS. Demo_M1.dmg runs on new computers with Apple Silicon (M1, M2 processors). Demo_X86_64.dmg runs on computers with the older Intel processors. You should simply be able to download, run the installer, and then run the app. Updates will be supplied as new installers. The installers may be found on the Sakai site, under the folder for "The language of electrical Signaling…". Because the program is not "signed", you may need to go to the System Preferences, Privacy settings, to allow it to run. This should only need to be done once.

For Linux or advanced macOS users:

If you want to work from the source code or are in Linux, you can download the source from https://github.com/pbmanis/neurodemo and set it up directly. Use the "main" branch.

1. Download Python for your operating system from https://www.python.org, version 3.10.6 or 3.10.7. If you already have Python installed, we recommend that you still install one of these later versions.
2. If you don't already have it, download "git" from https://git-scm.com/downloads for your operating system, and install it.
3. Decide where you want to save the simulator program ("demo"). One good location is on your desktop, inside a "Python" directory. Create the directory if it does not exist.
4. Run a bash or zsh terminal. In the terminal:
    a. "`cd ~home/Desktop/Python`" (or wherever you want to put this – it must be a place in the filesystem where you have write permission, however).
    b. Get the repository: "`git clone https://github.com/pbmanis/neurodemo`"
    c. go into the neurodemo directory ("`cd neurodemo`").
    d. Check to see that you are in the "main" branch (this is important!) This can appear as "(main)" before the command prompt. It should be the default branch after you do step b. If it is not, type: "`git checkout main`" at the prompt.
    e. Now you can run the install script "./make_local_env.sh". This will print out a lot of information as it proceeds, just be sure that it runs completely and doesn't have errors. If it is finishes successfully, the command prompt will be preceded by "`(neurodemo_venv)`" which indicates that a working environment has been created. . Make sure you are on the right branch ("main").
    f. Now you should be able to type "`python demo.py`" and it will bring the program up. If it comes up and looks like *Figure 1*, go ahead and close it (use the top bar to close).

g. Consider making a desktop shortcut to start the program, by activating the environment and running the demo.py script. Make sure you have the environment set (if it is not, try "./neurodemo_venv/bin/activate").
h. **Updates:** If we have an update or bug fix (very likely!), you should only need to go to the folder where you put the repository and type "`git pull`". Restarting the program should now bring up the new version; you do not need to go through the rest of the steps unless we have incorporated an additional requirement for the program that requires rebuilding the environment. We will let you know if that happens.

Interacting with the Simulator

Bring up the program by clicking on the icon () or starting the program from the terminal. You should see a new window that looks like Figure 1.

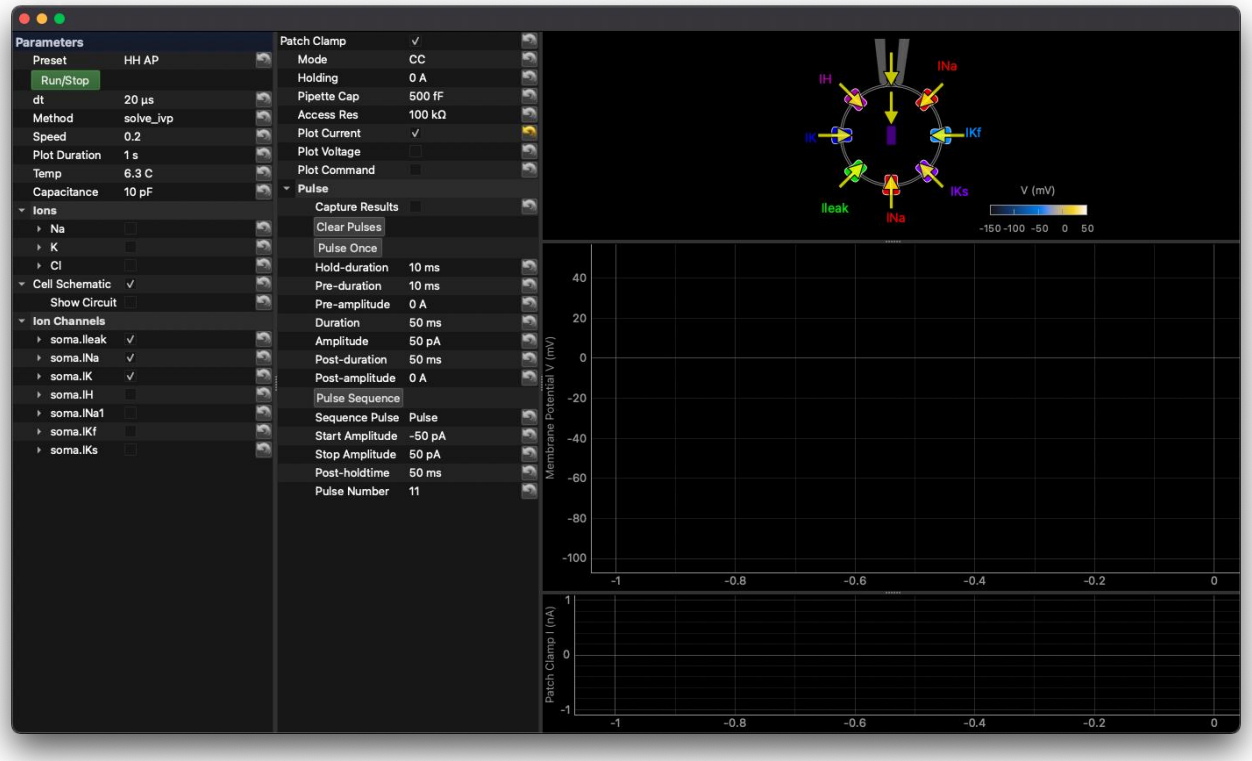


Figure 1 Neurodemo main screen

On the left is a panel that controls the **Parameters** of the simulations, with the parameters laid out in a hierarchical organization. The main subdivisions of this panel are: **Preset, Cell Schematic, and Ion Channels**. Note that all parameters except the speed, and counts (Pulse Number) have units.

In the middle is a panel that controls the Patch Clamp recording (top group) and Pulse generation (stimulation) (bottom group) parameters.

On the right are three panels showing graphics and plots.

1. The top shows the cell, the currently installed ion channels, and a patch clamp electrode. This can be switched to show the equivalent electrical circuit superimposed on the cell by checking the "Show Circuit" checkbox under "Cell Schematic."
2. In the middle is a plot of the membrane potential (when in Current Clamp) or electrode current (when in Voltage Clamp).
3. The lower panel shows the current command in current clamp and the voltage steps in the voltage clamp. Additional plots can be shown, as will be described later.

The panels can all be resized by clicking and dragging the splitter lines (grey) between them.

The two lower plotting panels have a set of controls built into them (these use the open-source graphics package "pyqtgraph," originally developed by Luke Campagnola and now used worldwide for fast, interactive graphics; see www.pyqtgraph.org). This package focuses on interaction with the data and on rapid display; however, it has a very rich feature set and can also be used to make publication-quality figures for papers.

The fundamental interactions with the graphs are:

1. To change the scale, place the mouse over the axis labels (numbers) and scroll; to slide the axis, hold down the left button while moving the mouse. If you place the mouse inside the plot, these actions affect both axes simultaneously to zoom in and out around the location of the pointer.
2. Right-clicking with the mouse will bring up a context menu that allows you to:
    a. Reset the view to show everything in the plot.
    b. Manually set the x and y axis limits.
    c. Change the mouse mode from "3-button" (typical for Windows and Linux) to "1-button" (Apple/macOS). Everything works the same in the 1-button mode except that when you are over a plot, pressing and dragging creates a transparent yellow rectangle that shows the new sized area that will be displayed when you release the button. This can be useful for closely examining specific details of the plot.
    d. Perform some simple calculations.
    e. Export the data in the plot to either a graph (in PNG, TIFF, or SVG format), a matplotlib figure or a CSV file that can be read into your favorite plotting program or Excel. In some of the problem sets and possibly on the exam, this will be needed to save the images of your simulations to be included in your answers.
3. Moving the mouse outside the graph axes, or towards the lower left corner, will reveal a little boxed letter "A." Clicking on this box will rescale the plot to show all the data. This is useful if you have zoomed in or out too far and need to reset to see things.

When you start a simulation (with the "Pulse" or "Pulse Sequence" buttons" while in "Run mode"), the traces will continuously scroll from right to left, showing the simulation results. When you run a "Pulse Sequence", a second window will pop up showing the responses to the sequence with superimposed traces. The simulator can run continuously, but you may wish to stop it with the Run/Stop button after a simulation.

Next, we describe the elements of each panel.

The Parameters panel

Parameters

This section controls some general parameters of the simulations.

- Preset: This drop down list provides some preconfigured settings for the simulations. Currently, this includes "HH AP": Hodgkin-Huxley action potential, "Passive": A passive cells with only leak channels and the membrane capacitance, and "LG AP": the Lewis-Gerstner model of a cortical neuron action potential.
- Run/Stop: This button starts and stops the simulator. It will be green when the simulator is NOT running, and pressing it will start the simulator. It will be red when the simulator is running, indicating that pressing it will stop the simulator.
- "dt": This controls the integration time step for the simulator. If the simulator seems to oscillate, you can decrease the interval to 50 or 20µs.
- Speed: This controls the display speed. A value of 1 is the nominal speed; smaller values will slow the simulations, and larger values will cause the scrolling display to run faster.
- Plot Duration: This controls the time axis of the scrolling graphs on the right. Typically, you will want this to be 1-3 seconds.
- Temp: Temperature for the simulations, in degrees Celsius. Normally, this would not be changed.
- Capacitance: The total capacitance of the cell membrane, in Farads (or pF, µF, nF).
- Many of these parameters have limits, so if it seems that you cannot change the value, you have probably moved outside the limits.

Ions:

This section controls the ion concentrations. When the Na value is NOT clicked, a standard set of equilibration potentials are used in the simulations. When clicked, the concentrations specified for Na, K and Cl are used for the simulations instead. Clicking on the triangle to the left of the label opens the list of parameters that can be changed:

- C[out] is the external ion concentration (in mM)
- C[in] is the internal ion concentration (in mM)

The Erev value is calculated and displayed for the ion at the selected temperature (as set by the Temp value). Erev is not editable.

- Cell Schematic: When this is checked, the cell schematic is shown in the right top graphics window. The schematic shows the cell, the pipette (top), a set of channels with different colors, and arrows that correspond to the magnitude and direction of current flow through the different elements. The schematic will change when you first run a simulation to only show the relevant channels and currents.
  - Show Circuit: Checking this box will switch to showing the equivalent electrical circuit for the channels and capacitances in the simulation, and the channel display will be partially hidden.

Ion Channels:

This section has controls for each of the types of ion channels in the simulators. Subsets of these will be selected according to the Preset in the Parameters section. Each channel can be included or excluded with the checkbox next to it's name. For brevity, here we describe the control parameters for one channel, but all channels have the same set of parameters:

- Gmax : The maximal conductance of the channel, in nS (1/giga-ohm).
- Erev : the reversal (or Nernst, if the Ions block is being used) potential for the channel.
- Plot I: This check box will add a plot of the current through the channel on the right.
- Plot G: This check box will add a plot of the conductance of the channel on the right.
- Plot OP: This checkbox will add a plot of the channel open probability on the right.
- There may be additional checkboxes listed here. These are the state variables of a model. For example, for the Hodgkin-Huxley model sodium channel, there are boxes for the activation gating variable, "m" and the inactivation gating variable "h". Together with the other plots, these can give you insight into how the gating variables relate to conductances and currents in the cell.

The Patch Clamp panel

This panel controls the virtual patch clamp amplifier used to run the simulations. The amplifier controls include:

- Mode: Current-clamp (inject current, measure voltage), or Voltage Clamp (Command voltage to the pipetted, measure current).
- Holding: This is the holding current (in current clamp) or voltage (in voltage clamp). The "holding" current is a current that is injected into the cell while no pulses are being generated.
- Pipette Cap: This is the capacitance of the pipette itself.
- Access Res: This is the resistance of the pipetted (also known as series resistance).
- Plot Current: A checkbox that adds a plot of the current. The current in current clamp is shown as the *sum* of the currents across the cell membrane, including the channels and the electrode.
- Plot Voltage: A checkbox that adds a plot of the membrane voltage in voltage clamp. This voltage is the voltage at the cell, not the voltage that was commanded to the clamp, and thus is affected by the membrane resistance and pipette capacitance.

Pulse:

The pulse parameters are controlled here. There are two buttons, “Pulse Once” and “Pulse Sequence”. Pulse Once generates a single pulse with the specified parameters. The pulses consist of 4 different phases. We will initially only use the primary pulse, but later will explore channel behavior using the pre- and post-pulses.

- holding, which is set by the Hold-duration and the Holding value in the Patch Clamp panel
- Pre-pulse: a pulse that is given before the Primary Pulse, with a given duration and amplitude.
- Primary pulse: The main pulse, which is set by the Duration and Amplitude values.
- Post-pulse: A pulse given after the Primary Pulse with the specified duration and amplitude.

Pulse Sequence:

This button initiates a run with a sequence of different pulse values. The parameters are:

- Sequence Pulse: This is a drop-down list where you select which phase of a pulse is sequenced: Pre, Primary or Post.
- Start Amplitude: The value of the first pulse in the sequence.
- Stop Amplitude: The value of the last pulse in the sequence.
- Pulse Number: The number of pulses in the sequence. If you make the start and stop similar, but opposite values, then using an odd number here will result in an evenly-spaced set of pulses in which one is at 0.
- Post-holdtime: This is the time in-between pulses in the sequence – that is, how long after the post-pulse is done before the next pulse begins with its holding phase.

You will notice that for many of the parameters there is a  symbol (it may be grey or yellow). When it is yellow, it means that the parameters have been changed from their original default values. Clicking the “return” button resets the parameters to default. If it is grey, the parameter is at its default value and the button does nothing. The symbol may appear also after changing the Preset. You do not need to manually reset the parameters after applying a Preset.

The units of the numbers are very important in these simulations. If you enter them incorrectly, it is possible for the running integrator to get “lost in space” and never return. For example, setting a stimulus current pulse to amperes (instead of pA), or a voltage step to 10 V instead of 10 mV, may cause this. To avoid this, you can stop the simulator when changing parameters, checking to see that they are correct before resuming. However, if you do cause the integrator to get lost, just close and restart the program.

New feature (9/20/2022): If you stop a run with some interesting features in a trace (such as action potentials), hovering the mouse over the trace will also cause the schematic to show the current magnitudes and direction at that time. If, while hovering over the traces, a vertical line is not following your cursor, update the program.

The Analysis Window

When you run a pulse sequence, a second window will appear, which looks like Figure 2. The pulse sequence window has 4 parts. The top panel shows the traces and the currents, as they are simulated. There is a “Hold data” checkbox, which when checked causes the traces to accumulate during simulations. On the right is a “Clear data” button, which will clear the display.

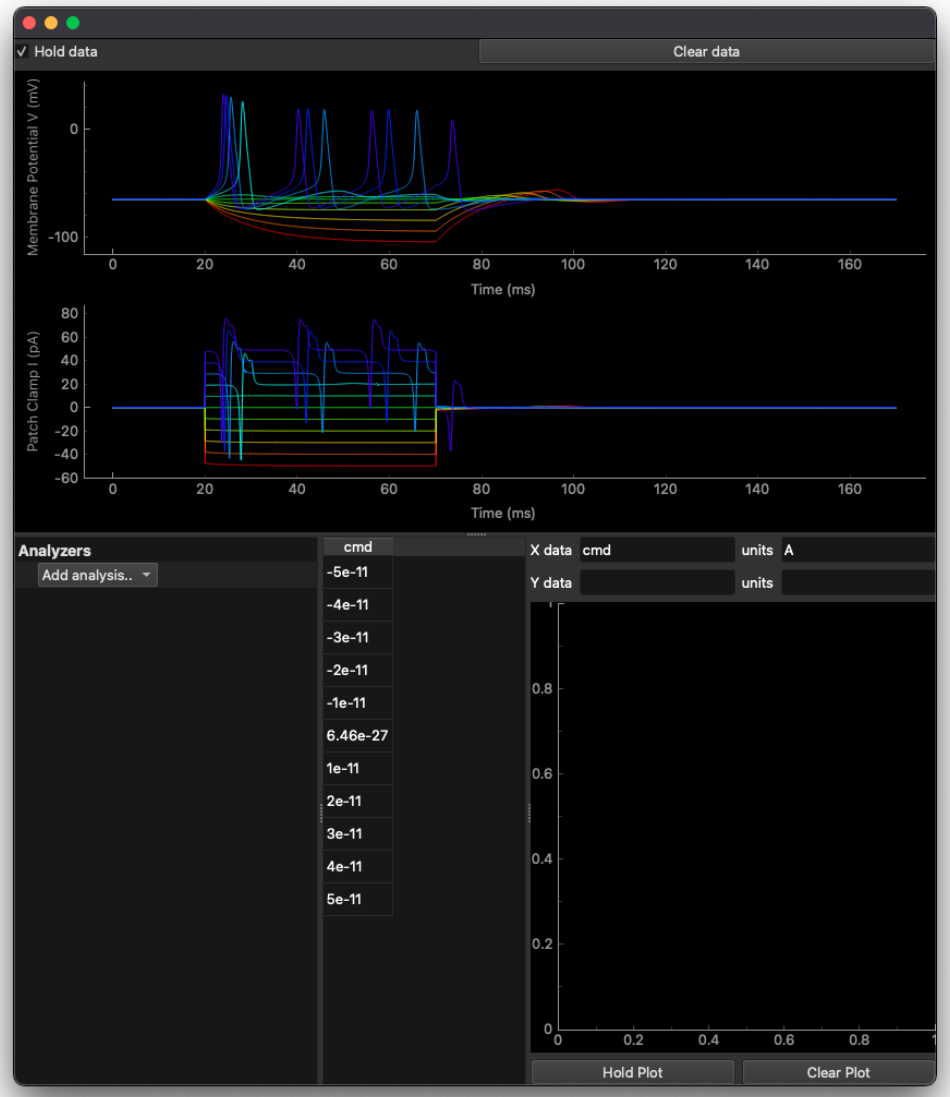


Figure 2 - Analysis Window

The lower half of the window has 3 parts: Analyzers, a table of values, and a tool for plotting calculations based on the values in the *cmd* table. The analyzer lets you select what will be measured, how it will be measured, and when it will be measured. You can have multiple analyses. The plot lets you plot various analyses, including some calculations based on different analysis results. We will walk through how the analysis tools works when we start the exercises. You can close this window if you do not need it; it will pop up again when you run another Pulse Sequence.

Exercise 1: The Passive Membrane.

Before we consider the roles of ion channels, we will first examine the “passive” properties of the cell membrane – those properties that depend only on the membrane resistance and capacitance. Let’s explore this with the program. Select the “Passive Membrane” from the Preset list and check “Plot Current” from the Patch Clamp pane. Start the simulator. You may want to set the speed to 1 and let it run for a few seconds, then stop it. Open up the soma.leak group under Ion Channels. The main window will not look like Figure 3.

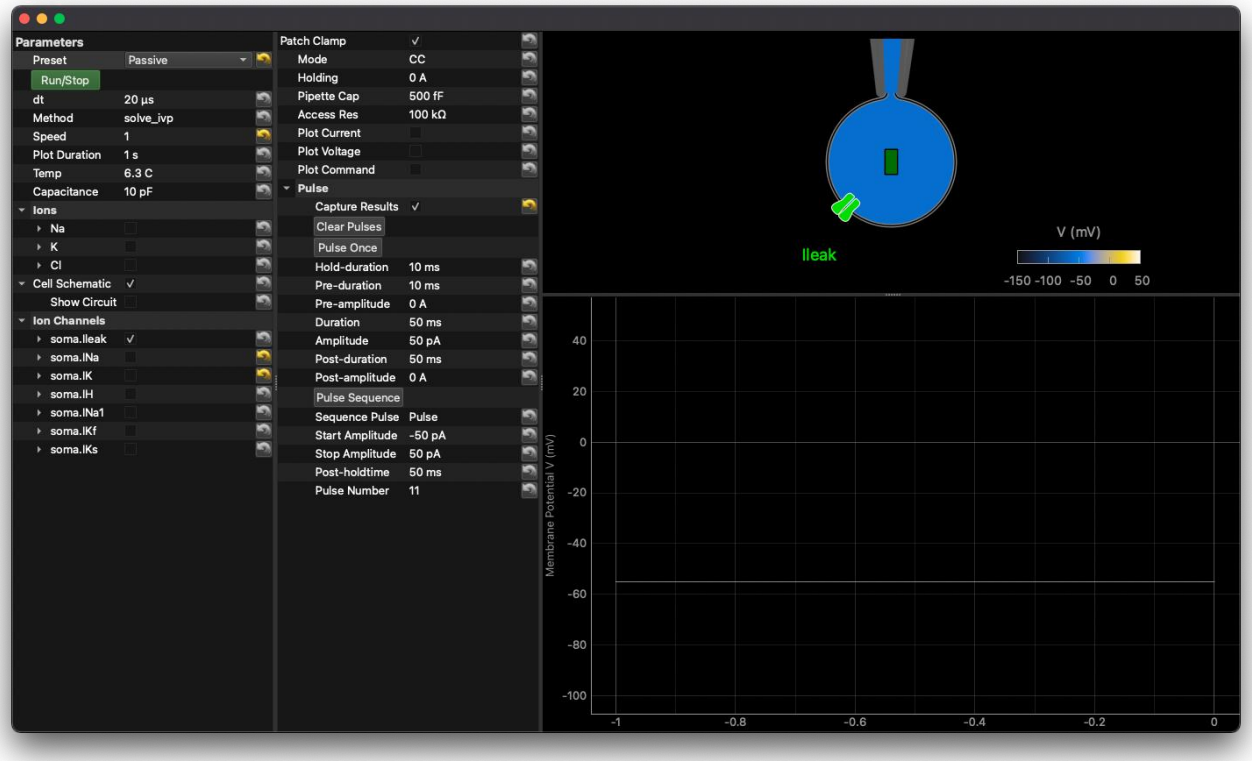


Figure 3 Passive cell model

The leak channel (green) is present, and the membrane potential is at -55 mV (Erev for the leak channel. (The cell is a sky blue, which corresponds to -55 mV).

Current-Clamp

There are two principal modes that are used for recording the electrical properties of cells: current-clamp and voltage-clamp. Let us first consider current-clamp. In this mode, we have an electrode attached or inserted into a cell, and pass current through the electrode to perturb the cell. It is important to note that in this mode, only the current through the electrode is actually “clamped”; the total current through the cell membrane can be doing something completely different, because it includes currents through ion channels and the membrane capacitance. Nonetheless current-clamp is frequently used because it is technically easy, and allows you to see the “natural” voltage changes that occur in a cell in response to injected currents or synaptic events.

Now, let’s inject a rectangular current pulse into the cell and see how the voltage changes. To help capture this, change the speed to 0.2, and on the right, set the Pulse Duration to 100 ms. Press “Pulse Once”, and then start the simulator. Be ready to stop it before the voltage change scrolls off the left side of the plot. Adjust the

plot scales to better show the pulses (click near the axes numbers and scroll – the cursor location is at the center of the region that is rescaled). Now you will see the voltage across the cell membrane Figure 4.

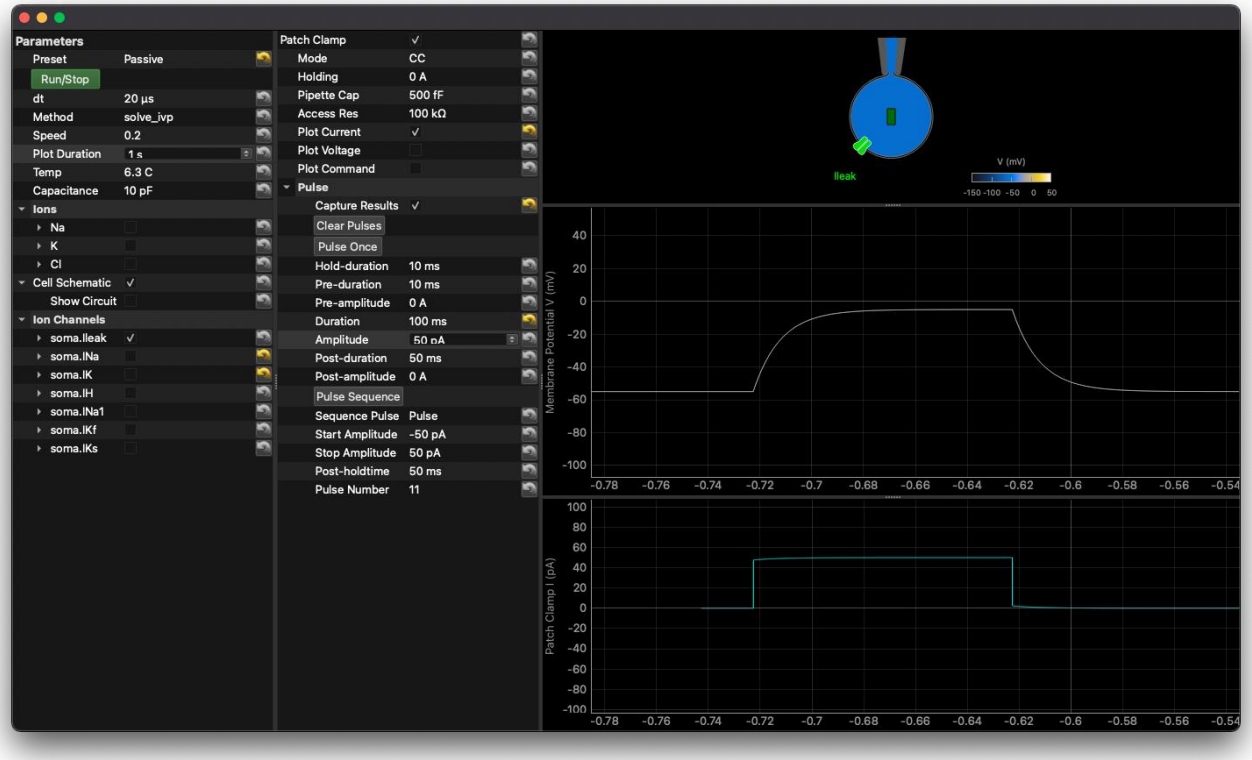


Figure 4 Passive cell model current pulse response

What you see is that the cell is at -55 mV before the pulse, and when the pulse steps up, the cell potential gradually climbs to around -5 mV. At the end of the pulse, the voltage falls back to the original potential with the same time course as the rise.

Let's explore what is happening here a bit more. Change the cell Capacitance to 50 pF and run the simulation again. Oops, something went wrong! The "oscillations" arise because the time steps were too slow in the solver. Change the "dt" (in Parameters) to 10 µs and try again. The voltage change is much slower (and may not fit in the 1-second window). Change the Capacitance to 2 pF and run again. Now the voltage change is faster.

> You may find it more intuitive and less confusing if you select "Plot Command", and unselect "Plot Current" in the Patch Clamp panel. "Plot Current" actually plots the sum of all the channel and electrode currents through the cell membrane, and so shows a measurement that you cannot actually make. You can also plot the capacitance currents separately, with the checkbox under the Capacitance value

What you observe here is that the membrane potential charges up with the current that is injected through the electrode. The rate at which the cell membrane charges depends on the total capacitance, as well as the membrane resistance (conductance). Under “Cell Schematic”, click “Show Circuit” (see Figure 5). Now the equivalent electrical circuit of the cell is shown. This includes the resistance of the leak, the “battery” that represents the Nernst potential of the leak, the transmembrane capacitance, and part of the circuit of the electrode (which we can ignore for now). All of the “wires” that go to the outside of the cell are connected together as “ground”, to complete the circuit.

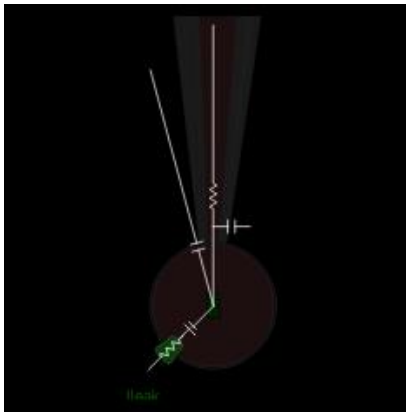


Figure 5 Passive cell model electrical schematic

Now, let’s do a bit of analysis on this passive cell. In the middle panel, under “Pulse”, check the “Capture Results” checkbox, press “Pulse Once”, and then press “Run/Stop”. A new window will pop up; once this happens, you can hit the “Run/Stop” button to stop the simulator. The window will look like Figure 6.

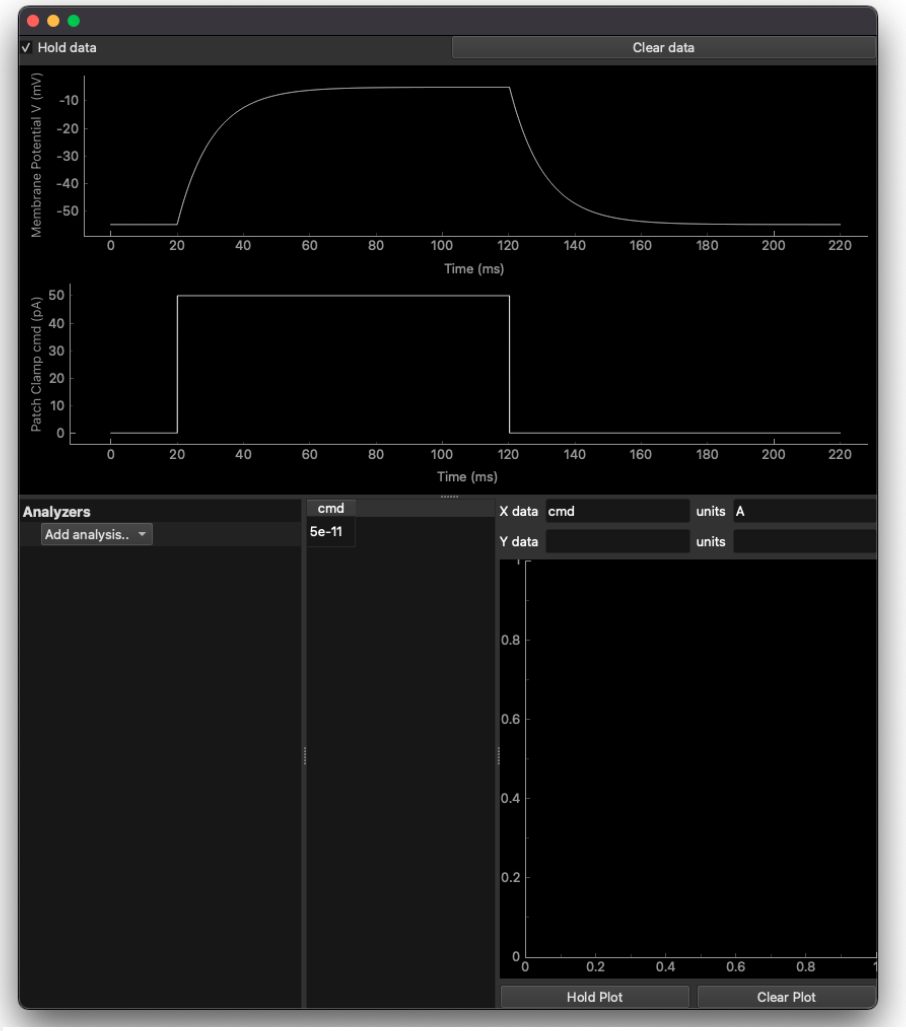


Figure 6 Analyzer window

The top two traces are the voltage and current. (The current is the total current through the cell, including the electrode current AND any channel current). Below this is an analysis section with 3 panes. The leftmost pane allows you to build “Analyzers”. The middle panel is a table of results. The right panel is a summary X-Y plot.

Under Analyzers, in the drop-down list “Add Analysis”, select “exp_tau”. The “exp_tau” tool has an Input (currently, “soma.V”, the soma voltage), a type of analysis (in this case, “expT_tau”), and Start and End times (See Figure 7). When you create an Analyzer, a pair of vertical yellow lines will appear in the top plot, with a dark blue transparent region between them. The lines can individually be dragged to select the start and

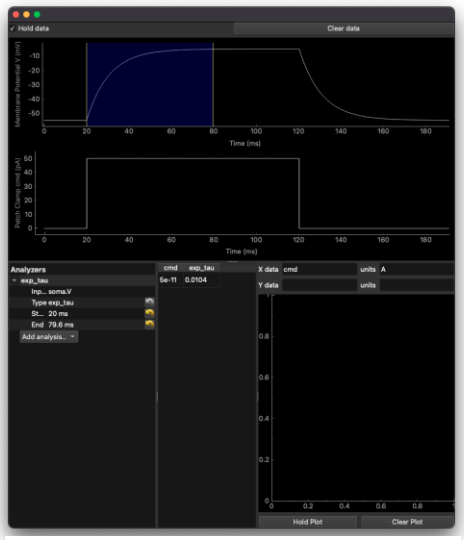


Figure 7 Measuring the membrane time constant.

end times in the analysis. The blue area can be clicked and dragged to slide the region along the time axis. Note that there is also a new column in the table, called "exp_tau", with a number, which is the decay time constant in ms. If you adjust the region to go from 20 to 80 ms (see below), you will see that the table value changes to 0.010x (x depends on exactly how you set the cursors), which is the decay time constant in seconds.

Now, let us think about this. We have a cell with 10 pF capacitance, and 1Gohm resistance (1/1nS conductance). Remember that the membrane time constant is the product of R and C: $(\tau\ (\mathrm{s}) = \mathrm{R}\ (\Omega)\ \times \mathrm{C}\ (\mathrm{F}))$. So, we expect $(1 \times 10^{9}\ \mathrm{W})\ \times\ (10 \times 10^{-12}\ F) = 10\ \times\ 10^{-3}\ s\ =\ 0.010\ s\ or\ 10\ ms.$

Is the time constant the same at the end of the step? Try it!

> It is important that the start of the selected region be after the start of the change in the current step; if it is before this then the fitter includes some of the flat part of the trace and the fit will not be accurate. The fit region must also include the asymptotic part of the change.

There is a very small deviation from the exact value of 10 ms. This arises from the Pipette capacitance of 0.5 pF (in the Patch Clamp parameters), which is in parallel with the membrane capacitance and therefore adds to the circuit. If you lower this value to 0.1 pF (the smallest allowed by the program), and re-run the pulse, you will see that you can get a value closer to, but not exactly 10 ms. If you greatly expand the traces in the top plot of the analyzer, you will in fact see that there are 2 traces that are just barely different.

We will not be concerned with the exact values in these simulations, but this observation raises an important practical point: *The properties of the recording electrode are a part of the system that you are measuring and affect the results that you get.* Most amplifiers that are used for electrophysiology have incorporated approaches for compensating for the electrical properties of the electrode and applying these can be important for obtaining reproducible and accurate results.

Questions

a. If you increase the cell capacitance to 25 pF, what is the time constant?
b. If you change the value for E_{leak} (in the Ion Channels, soma.ILeak parameters) from -55 to -75 mV, would you think that this changes the membrane time constant? Write down your answer and reasoning. Then, measure it to find out.
c.

Voltage-clamp

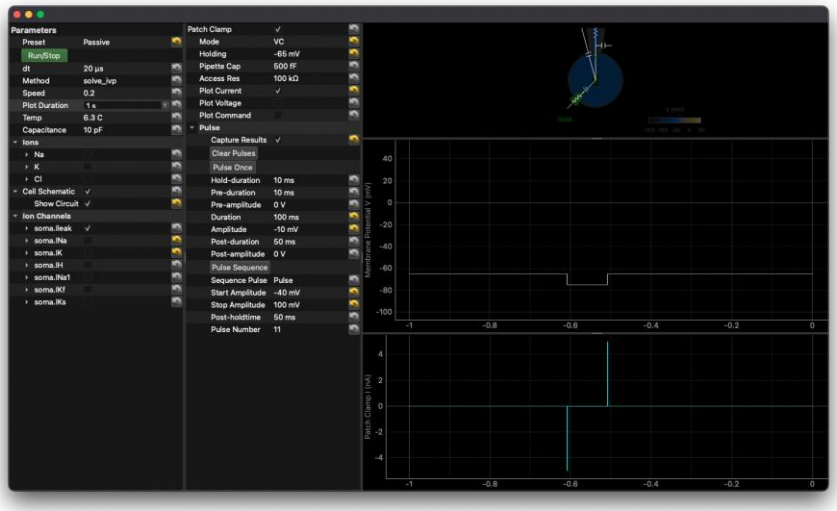


Figure 8 Voltage clamp setup and clamp current in response to a voltage step.

As we mentioned above, another method is "voltage-clamp". This approach was introduced in the 1940's by some clever engineers who were studying the mechanisms of how action potentials were generated[1] . This method strives to control the membrane voltage while allowing the measurement of how much current is needed to accomplish this. Voltage-clamp, together with pharmacology, allows the separation of different kinds of currents through the cell membrane, the measurement of ion channel opening and closing rates, and even the measurement of the currents through single ion channels (e.g., a single membrane protein). The development of the voltage-clamp method revolutionized electrophysiology of neurons and muscles. The subsequent development of the "patch-clamp" approach took this one step further, allowing the application of the method to smaller cells, and reducing the noise level of recordings. Voltage-clamp is not without technical problems however, and some of these can significantly affect the quality and interpretation of the measurements. It is important to be aware of these issues and to reflect on how they might influence the results reported in the literature.

Let's begin with considering voltage-clamp of the simple passive cell membrane that we just studied above. Close the analyzer window. In the main window, under Patch Clamp, change the "Mode" to VC (for Voltage Clamp). You will note that the pulse amplitudes now have units of V (or mV) instead of A (or nA or pA). Change the "dt" to 20 µs if it is not already at that value. Press "Run/Stop", and let the program run until a pulse has been generated, then stop the simulator. Press the "Clear Data" button at the top of the Analyzer window. If you left "Capture Results" checked, then the analyzer window will show the traces.

The main window will look like Figure 8, and the Analyzer window will look like Figure 9. Here, we have stepped the membrane voltage from -65 to -75 mV for 100 ms, then returned to -65 mV. In the bottom, we have the currents that were needed to keep the cell membrane at these voltages. Note that the current has two "spikes", which are associated with the charging of the membrane capacitance. If you expand these, you will see that they are only 1 point wide This is a limitation of the simulator, which only goes down to 20 µs steps. The time constant for completing the charging of the membrane is 10pF x 10^5 MΩ, or 1 µs. If you change the "Access Res" in the Patch Clamp panel to

[1] For a diversion, read Alan Hodgkin and Andrew Huxley's autobiographies in the Society for Neuroscience History of Neuroscience at https://www.sfn.org/about/history-of-neuroscience/autobiographical-chapters.

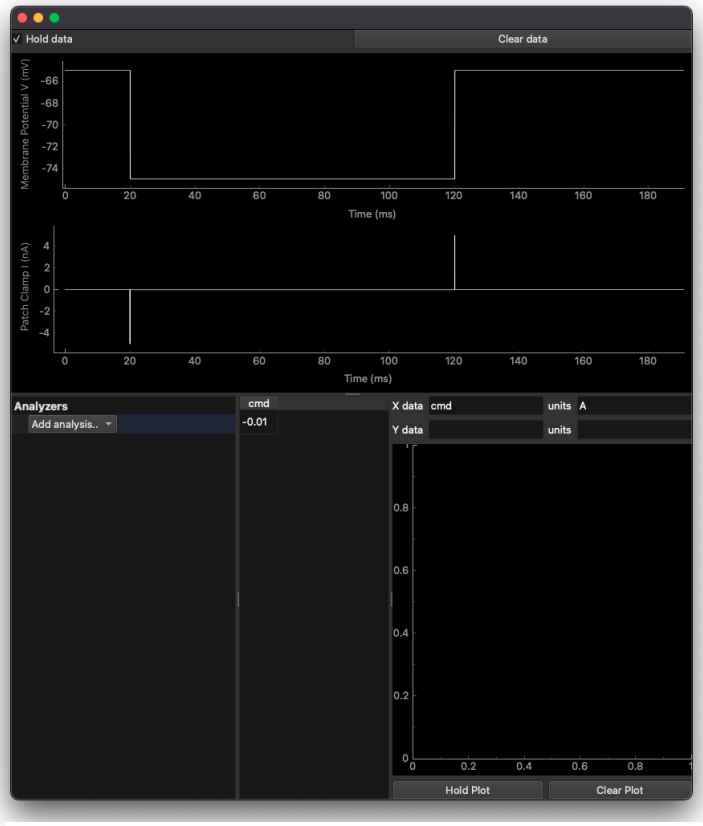


Figure 9 The voltage step is in the top panel, and the current transients associated with charging the cell capacitance are in the middle panel.

5 MΩ, this charging will lengthen to 50 µs. Running the simulation will reveal that the transients are smaller and slower. You will also note that the voltage across the membrane takes about 200 µs to reach the command value, even though the actual command step went immediately from -65 to -75 mV. This is because the current through the membrane went into charging the membrane capacitance but is limited by the access resistance. Once the capacitance has been charged, the only current needed to keep the voltage at -75 mV is that going through the resistance. Because the resistance is very high (1GΩ), the current is tiny. You can calculate that it should be: I = g*(V-V_r) where g is 1 ns (1/1GΩ), V is -75 mV, and V_r is -55 mV: 1*10-9 * (-0.075V – (-0.055V)) = $-2*10^{-11}$, or 20 pA.

We will explore using the voltage-clamp mode to examine how ion channels work in the next section. For now, all you need to know is that these capacitance transients will be present in the simulations (and in real recordings!). They can be shortened by reducing the access resistance and made smaller by having a smaller membrane area and by reducing the pipette capacitance.

Exercise 2: The Action Potential

In this exercise, we will examine the action potential. To set up for this set of exercises, close and restart the simulator, making sure that "HH AP" is selected in the Preset drop-down list.

The action potential is generated by the interaction of two different kinds of ion channels: sodium channels and potassium channels. In the simulation here, the sodium and potassium channels are based on the original Hodgkin-Huxley channels, so the traces that you observe and the measurements that you make should be comparable to their classic recordings.

The equation that is solved is based on:

$$\mathrm{I} = \mathrm{C_m} \frac{\mathrm{dV}}{\mathrm{dt}}$$

Where we compute I from the sum of all of the different currents that are present in the cell (and rearranging):

$$\frac{\mathrm{dV}}{\mathrm{dt}} = -\frac{1}{\mathrm{C_m}} \left(\mathrm{g_{Na}}(\mathrm{V} - \mathrm{E_{Na}}) + \mathrm{g_K}(\mathrm{V} - \mathrm{E_K}) + \mathrm{g_l}(\mathrm{V} - \mathrm{E_l})\right)$$

Ena, Ek and El are the Nernst potentials for Na K and the leak respectively. The currents for each class of channel are determined by their conductance and driving force, in the general form of:

$$I_x = g_x(V - E_x)$$

For each channel type *x*. This is just applying Ohm's law to compute the current for each channel type. If we have an electrode, then we have an additional current, I_{elec}, that we can add to the right side of the equation, inside the parentheses, as well. The solution to this equation is found by integrating both sides over time, which lets you solve for V. There are two considerations however that come into play. First, g_{Na} and g_K are also functions of time and voltage. In the Hodgkin-Huxley formulation, they are described through an intermediate set of parameters, called the "gating" variables, which represent the time evolution and voltage-dependence of the channels. These are typically called m and h for the Na^+channel, and n for the K^+ channel. The gating variables have their own formulation in terms of another set of differential equations and are sometimes specific to the channels being represented. We will discuss these more in a later section.

Obviously, the math for this can get a bit complicated very rapidly. However, there are some standard representations that are used that consider the opening and closing of channels as a simple first-order chemical reaction, and which also consider the stoichiometry of channel opening steps. We will discuss these more in connection with exercises where we examine the specific behavior of the sodium and potassium channels in isolation.

For now, let us just consider that the action potential is generated with the membrane is depolarized enough to cause a critical number of sodium channels to open, which leads to more sodium channels opening. This is the "sodium rushes in" phase of the action potential. As the membrane depolarizes (moves positive to the resting potentials) towards the sodium equilibrium potential, the potassium channels begin to open. Compared to the sodium channels, the potassium channels are a little slow and late (sometimes called "delayed rectifiers"), so they don't come into play right away, but when they do, potassium "rushes out", which drives the membrane potential back towards the potassium equilibrium potential. But that is not all –

the sodium channels have a property called “inactivation”, which causes them to close quickly, within a millisecond or so after they open, so they no longer pass a current that is trying to drive the cell to E_{Na}. The result is that the K^+ channels are free to drive the membrane potential all the way back to rest, and everything gets reset, ready to generate a new action potential.

Now, start the simulator. Hit “Run/Stop”. You should cell that the cell has 3 channels in it – IK, Ileak, and Ina. The small yellow arrows indicate the direction of current flow for each of the channels at the resting potential, which is about -65mV. With the simulator running hit “Pulse Once”. You will see a group of 4 action potentials slide by – probably too quickly to appreciate them. Check the “Capture Results” in the Pulse section and pulse the cell again – now you will see the 4 action potentials in a new window. You can expand the time scale to look at the action potentials more closely (place the mouse cursor just below the x axis line, click and slide “up”, or use a scroll wheel on the mouse). The top panel of the new window should look like this:

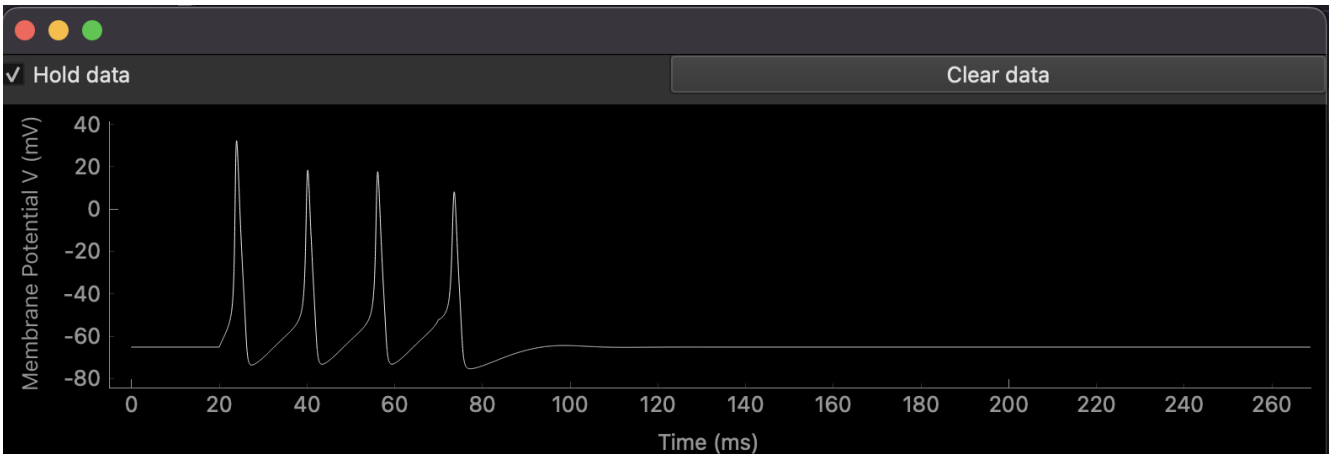


Figure 10 Action potentials

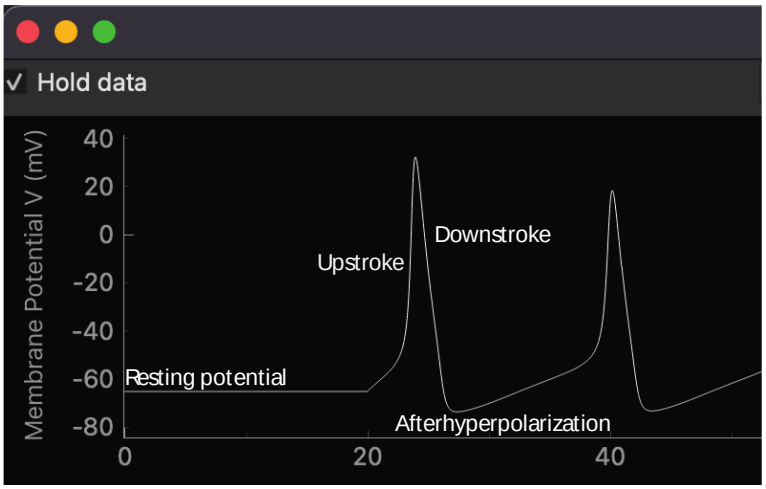


Figure 11 Action potential components

Now go ahead and explore different pulse sizes and durations to see how they affect the action potentials. You can control these parameters with the “Duration“ and “Amplitude” values in the Pulse panel. Remember to watch the units when you edit the values! You can always reset the values by clicking on the curved yellow arrow to the right of the editable value field.

Questions

a. Set the duration to 5 ms. What is the smallest current that you find can elicit an action potential? This value would be called the "rheobase" for a 5 msec long current pulse.
b. Set the current to 50 pA. What is the shortest duration pulse that will generate an action potential? This value is called the "chronaxie" for a 50 pA pulse.
c. There is a trade-off between the duration and the current amplitude. Try systematically varying the duration and looking for the smallest current that produces an action potential at each duration. Using the "Pulse Sequence" can speed this up considerably. The relationship between the current and the duration can be plotted and is called a "strength-duration" curve.
d. Try setting the pulse duration to a long value (such as 500 ms). Now vary the amplitude find out where you can get just one action potential, or 3-4 action potentials, or a train that lasts the duration of the pulse. When you get just one or two action potentials, what happens after the action potentials? Why do you think this happens?

How long do you need to wait to generate a second action potential?

Next, let's try looking at the refractory period. This is the time after an action potential where it is difficult to elicit a second action potential. There are two parts to the refractory period, typically called the "absolute" and "relative" refractory periods. During the "absolute" refractory period, a second action potential cannot be elicited regardless of how strong the stimulus is. During the relative refractory period, the stimulus must be stronger than the threshold stimulus from rest but can still elicit an action potential.

Try this:
Set the Pre-duration to 4 msec, the pre-amplitude to 30 pA, the Duration to 10 ms, the Amplitude to 0 ms, the post-duration to 4 ms (same as pre-duration), and the post-amplitude to 0 pA. Clear the data (not pulses) in the second "capture" window. Pulse once, and you should get one action potential.

We are now going to generate two stimuli, by using the "post amplitude"; the interval between the stimuli is set by the "Duration" parameter. Now set the post-amplitude to 30 pA. Pulse once and compare the two traces. The second stimulus did not make an action potential. For this 10 msec interval, what current level to you need to get a full action potential?

If you look at the responses to current levels just below the level that can generate an action potential, you will see that the shape suggests that the cell is trying to make an action potential but can't quite make get started.

Bonus: what happens if you make the Amplitude -20 pA? Why do you think this happens? Are you surprised? We will discuss the mechanism(s) for this more later!

Next, set the post-amplitude to 30 pA, the same value as the pre-amplitude (with the Amplitude set to 0). What is the shortest duration (explore by varying the Duration) that lets you generate 2 action potentials? If you change the pre and post amplitudes (keeping them the same for each trial), how does this duration change?

There is a trade-off between the stimulus amplitude and the time after an action potential during the relative refractory period in the ability to generate a second action potential with a short pulse.

Exercise 3: The Potassium Channel: Activation and Deactivation

Potassium channels may not be the most exciting channels, but they are extremely important and diverse, controlling many aspects of cell firing patterns, beyond just repolarizing action potentials. They are also targets of protein kinases and phosphatases as mechanisms to regulate the electrical excitability of a cell. Hodgkin and Huxley did the first kinetic characterization of potassium channels in squid axon, and their approach has been applied in many studies since. We will start with delayed rectifier potassium channels because they are relatively simple and are useful for understanding the first principles of how ion channels operate to control membrane potential, and how they are gated by membrane potential changes.

To do this, we are going to also introduce a few concepts about voltage-clamp recording, and how to interpret voltage-clamp recordings.

For the simulations in this section, close and restart the simulator, and make sure that the preset is "HH AP". In the "Ion Channels" section, uncheck "soma.Ina". This turns off the sodium currents. If you do this while the simulator is running, you will notice that there is a tiny hyperpolarizing shift in the resting potential. This is normal and occurs because, at rest, a tiny fraction of the sodium channels are open, which pushes the membrane potential slightly towards E_{Na}.

The experiments we are going to do here will be in "voltage clamp". This is a recording mode in which we attempt to control the membrane voltage, while recording the membrane current. The feat is accomplished by using a feedback circuit and a current-to-voltage convertor at the electrode. Lets try it – change the Mode in the Patch Clamp to VC. You will notice that the parameters in the Pulse section are now in mV instead of pA (or A). The values are *relative* to the holding potential, which is set in the Patch Clamp section to -65 mV. Check the "Capture Results", and then click on Pulse Once.

In the capture window, you will see the voltage step going from -65 to -75 mV. Below this in the Patch Clamp I trace, you will see two current spikes. By now you should know that these are the capacitive spikes associated with charging the cell membrane. We can make them smaller by properly "compensating" the amplifier as much as we can, so set the Pipette Cap to 100 fF, and the Access Res to 10 kΩ.

Now go ahead and run a Pulse Sequence. The captured traces should look like this:

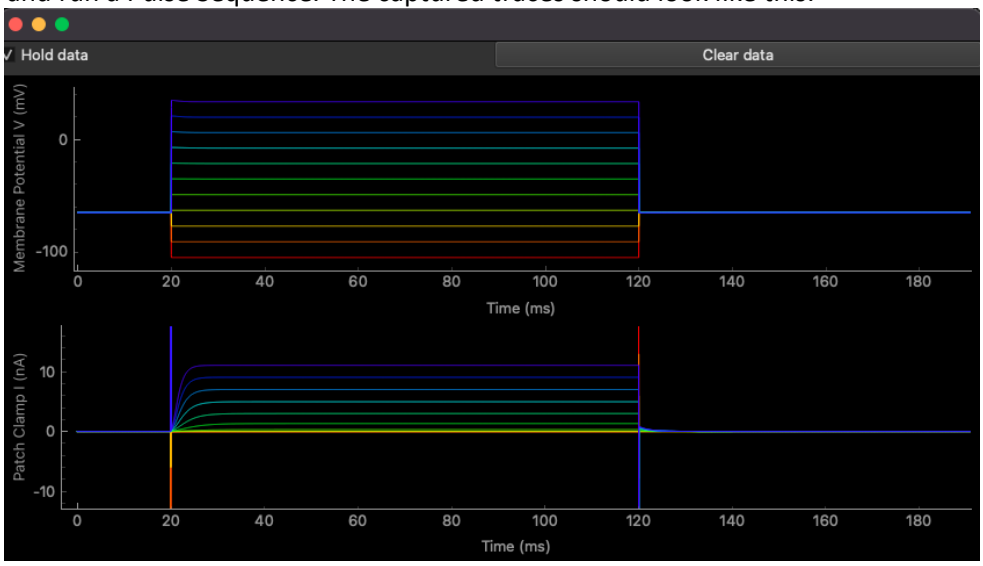


Figure 12 Voltage Clamp of Potassium channels

The traces are in different colors so that you can compare the voltage steps with the corresponding current steps.

The small capacity spikes are present at the start and end of the traces – you can ignore them, but know that this is what real data would look like. Starting at 20 msec, you see currents that rise over the next 10 ms, then stay constant until the end of the voltage step. These are the currents through the K channels. You don't see any currents for the red traces (that are below the holding potential), but only for the ones that are positive to the holding potential. This is because the channels are voltage-dependent, and do not open until the membrane is depolarized. At the end of the pulse, you also see some small currents that decay back to the baseline (after 70 ms). These are called "tail currents" and are an important tool in studying both voltage-dependent ion channels and even currents through neurotransmitter receptors. We will analyze these in more detail shortly.

Let's do a little analysis. In the Capture window, go to "Analyzers", and click on "Add analysis", and select "mean". In the "Input" field., change the drop-down value to "soma.PatchClamp.I" This will make a measurement of the average current in some time window. Adjust the window to end just before 70 ms. In the right pane, fill out the plot parameters, with Xdata "cmd-0.065", units "V", Ydata "mean", and units "A". you should get a plot similar to Figure 13. The equation "cmd-0.065" adds the holding potential from the top of the Patch Clamp panel to the command, so you are plotting the currents relative to the actual cell membrane potential. This is the current-voltage relationship of the potassium conductance.

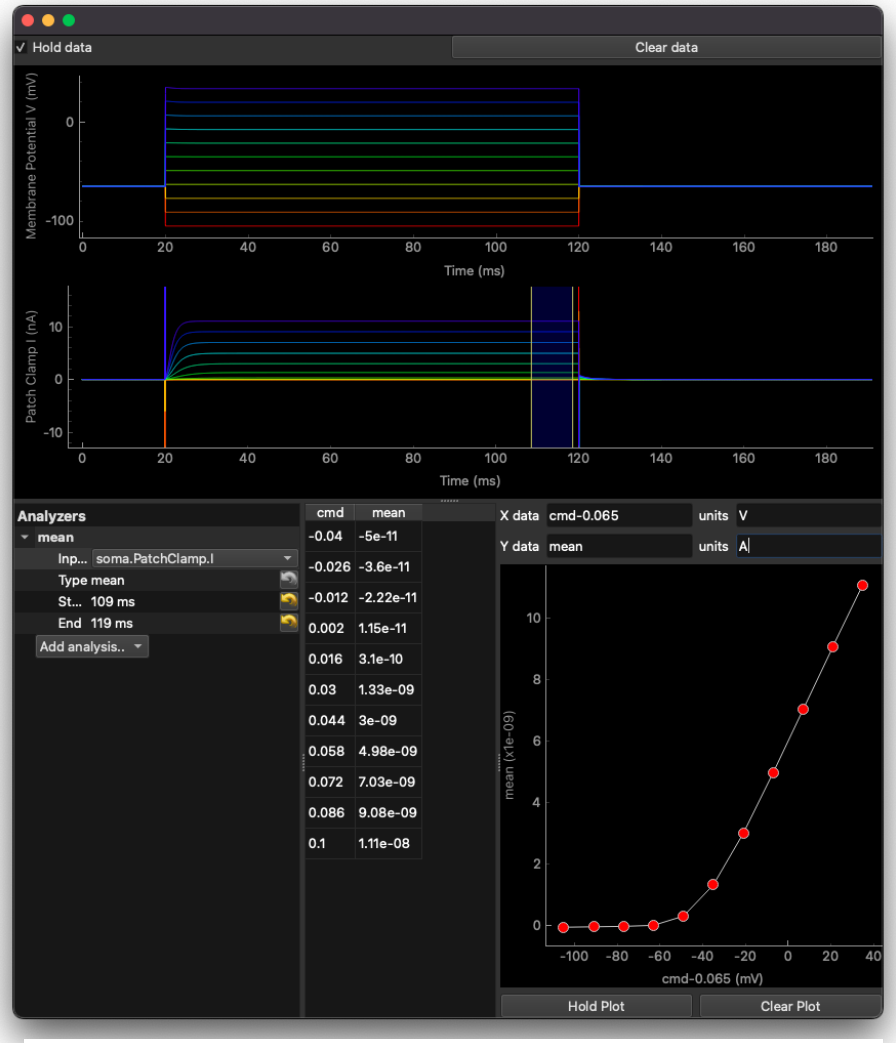


Figure 13 Steady State current-voltage relationship of the H-H delayed rectifier obtained by plotting the current in the lower traces against the voltage in the upper traces

Questions:

a. Why does the current continue to grow in a linear manner as the step voltage gets more positive?
b. Can you reduce the transient currents at the beginning and end of the step?
c. Try calculating the *conductance* from the current measurements. E_{rev} is -77 mV (open up the soma.IK parameters in the Ion Channel section).

d. Why is the maximal conductance about 100 nS in this plot? What should the actual value be (look at the soma.IK parameters)?

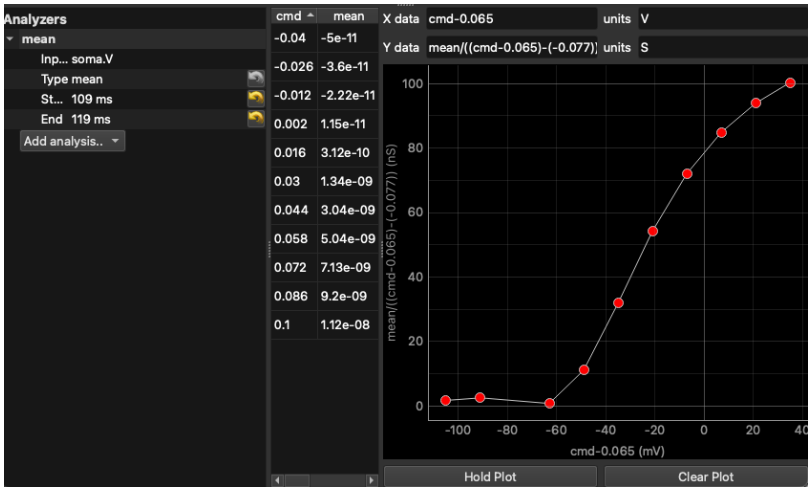


Figure 14 Conductance Plot from the current voltage-relationship in Figure 13. The equation in Y data is cut off, but reads: mean/((cmd-0.065)-(-0.077)). This is the mean current, divided by the difference between the command potential and the reversal.

Activation of the K^+channels

You can see that the currents at the start of the voltage step rise slowly and smoothly. If you look more carefully at the voltages, you will see that the rise is actually sigmoidal in shape, and that it seems to get faster with more depolarization. To see this more clearly, turn off "Plot Current" in the Patch Clamp pane, and check "Plot I" in the soma.IK parameters. Now we will plot *only* the K^+ currents, without contamination from the cell membrane capacitance. Clear the analysis plot, and re-run the sequence. To recalculate the conductance, you will need to insert a minus sign in front of the Y data equation to correct for a convention used in programming this plot.

Here is an expanded view of the activation of the currents:

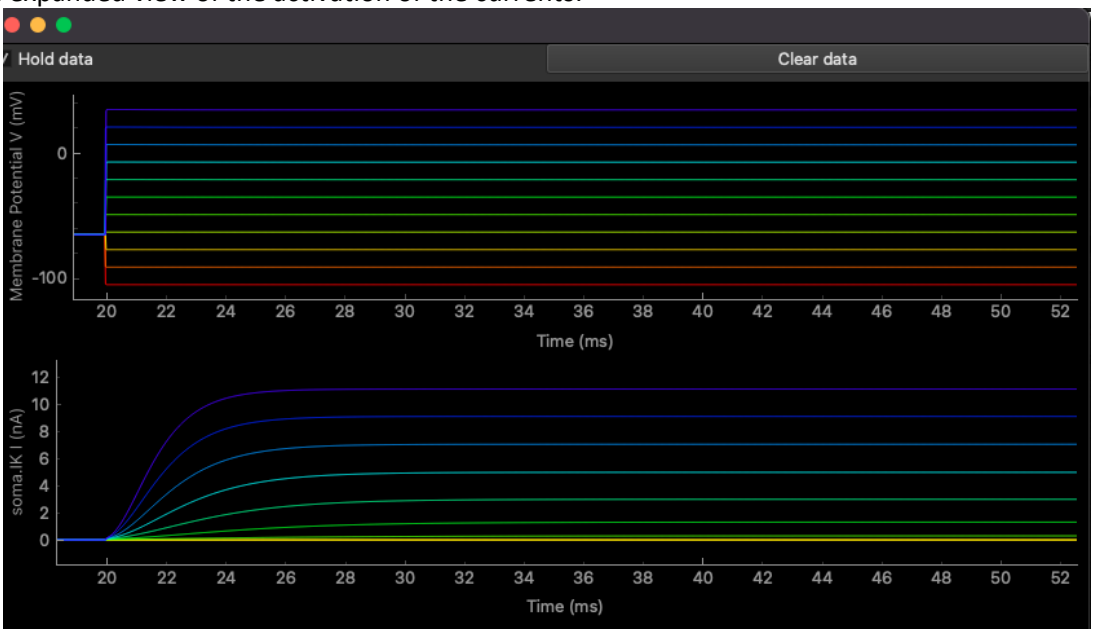


Figure 15 Close-up view of K channel activation

Here is Figure 2 from Hodgkin and Huxley, 1952c:

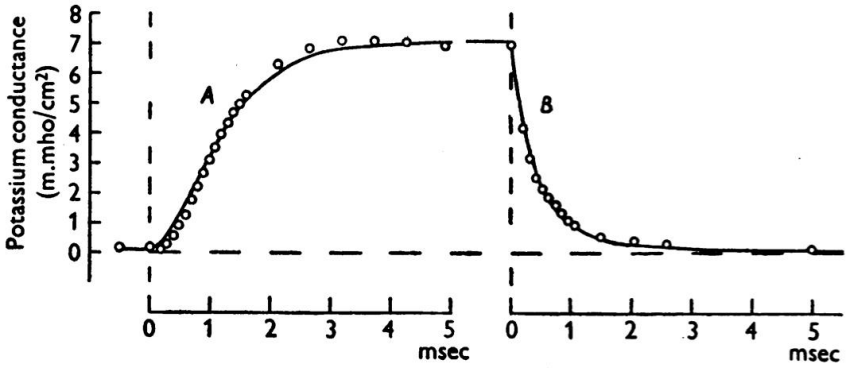


Fig. 2. *A*, rise of potassium conductance associated with depolarization of 25 mV; *B*, fall of potassium conductance associated with repolarization to the resting potential. Circles: experimental points replotted from Hodgkin & Huxley (1952*b*, Fig. 13). The last point of *A* is the same as the first point in *B*. Axon 18, 21° C in choline sea water. The smooth curve is drawn according to eqn. (11) with the following parameters:

	Curve *A* ($V = -25$ mV)	Curve *B* ($V = 0$)
g_{K0}	0·09 m.mho/cm²	7·06 m.mho/cm²
$g_{K\infty}$	7·06 m.mho/cm²	0·09 m.mho/cm²
τ_n	0·75 msec	1·1 msec

Figure 16 Figure 2 from Hodgkin and Huxley, 1952c

You can see that the rise in A is also sigmoidal. The points are the data, the line is calculated. The sigmoidal rise can be thought of as a requirement for cooperative gating of 4 energetically identical protein subunits that must all be in the correct conformation for the channel to pass ions. Only when all 4 are in the right conformation will currents be passed. This can be thought of as:

$$C0 \leftrightharpoons C1 \leftrightharpoons C2 \leftrightharpoons C3 \leftrightharpoons O4$$

Where each C represents a state with 0, 1, 2, or 3 subunits in the open conformation, and O4 represents the state with all subunits in the open conformation. This gives rise to a 4th order activation time course (4 sequential transitions), and hence "delayed rectifier".

One consequence of a cooperative interaction for gating is that when the voltage is then returned to a more negative level, any *one* of the subunits changing to the "closed" conformation is sufficient to close the channel. In the equation above, you can see that there is only a single step from O4 to C3. This is a first-order process, meaning that the closing of the channels follows a single exponential time course. Note that in Figure 16 (Figure 2 of HH 1952c), part B looks like a single exponential decay.

Questions

a. Now, using the simulator, look closely at the decay following the end of the pulse. Can you plot the decay time constants?
b. Why do you think the rise is sigmoidal? How might you plot the data to show that it is?

A bit about the gating equations and deriving the values from measurements.

For the squid axon K^+ channel in the Hodgkin-Huxley formulatoin, there is only one gating variable, which by convention is called "n". n is the fraction of channels that are open at any given time. In the formula for current through any kind of channel, x:

$$I_x = g_x(V - E_x)$$

Importantly, g_x includes both voltage-dependent ant time-dependent terms, as follows:

$$g_K = \bar{g}_K n^4$$

Where n is the gating variable, which ranges from 0 to 1, and $\bar{g}_K$ is the maximal (or total) potassium conductance in the membrane for this type of channel.

The time evolution of n is also represented as a differential equation:

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n$$

α_n and β_n are rates for the opening and closing of channels, which vary with voltage, but not time. We can consider first a simple channel with just two states:

$$C \leftrightharpoons O$$

With forward rates (from closed to open) of $\alpha(V)$ and reverse rates $\beta(V)$. These rates are usually an approximately exponential function of voltage (Figure 17). In fact, by measuring the rates at which the currents activate (channels open), and at which they close (using tail currents), at different voltages, one can plot the rate curves, and find an equation that fits them.

In response to a voltage step at time 0, the value of n will change exponentially in time as:

$$\mathrm{n} = n_\infty - \left(n_\infty - n_0\right) e^{\{-t/\tau_n)}$$

$$\text{with } n_\infty = \frac{\alpha_n}{(\alpha_n + \beta_n)}$$

$$\text{and } \tau_\mathrm{n} = \frac{1}{(\alpha_\mathrm{n} + \beta_\mathrm{n})}$$

n_∞ is the steady-state value at the voltage that is stepped to, and n_0 is the value of n prior to the step.

The equations for n_∞ and τ_n define one approach for experimentally extracting the rate functions. You can measure the steady-state activation, n_∞, in voltage clamp at different voltages with voltage steps, waiting for the current to reach steady-state. You can then measure the time constant for activation or deactivation at those voltages by fitting the activation or deactivation time courses with exponential (or exponential raised to a power) functions. Since the time constant is n_∞/α_n, knowing τ_n allows you to compute α_n from the data, and then knowing α_n and τ_n, you can calculate β_n.

Figure 17 shows this applied by Hodgkin and Huxley and shows how the rates vary with voltage.

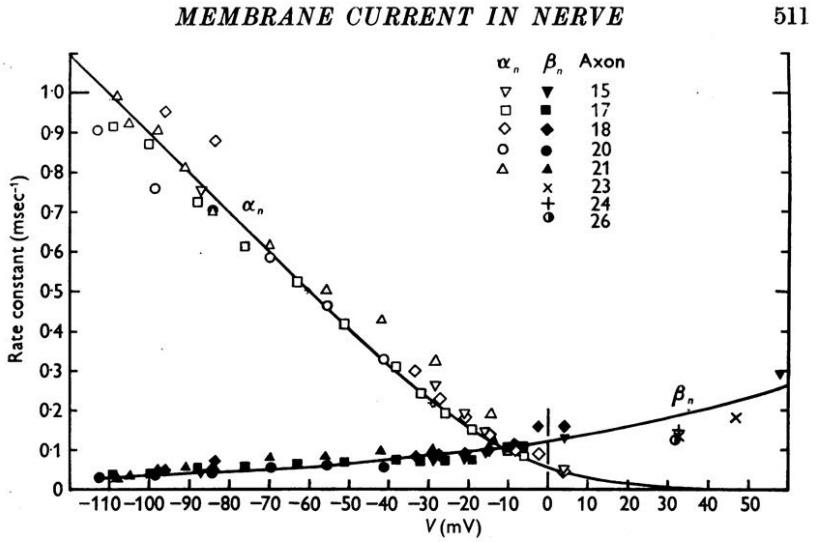


Fig. 4. Abscissa: membrane potential minus resting potential in sea water. Ordinate: rate constants determining rise (α_n) or fall (β_n) of potassium conductance at 6° C. The resting potential was assumed to be 4 mV higher in choline sea water than in ordinary sea water. Temperature differences were allowed for by assuming a Q_{10} of 3. All values for $V<0$ were obtained by the method illustrated by Fig. 3 and Table 1; those for $V>0$ were obtained from the decline of potassium conductance associated with an increase of membrane potential or from repolarization to the resting potential in choline sea water (e.g. Fig. 2). Axons 17–21 at 6–11° C, the remainder at about 20° C. The smooth curves were drawn from eqns. (12) and (13).

Figure 17 - Figure 4 from Hodgkin and Huxley, 1952c

These rate changes are typical, and if you calculate τ_n , you will find that it is roughly bell-shaped, with the longest time constants near the resting potential and much shorter time constants at more depolarized and hyperpolarized potentials.

Exercise 4: Activating, deactivating, and inactivating ion channels during an action potential

Of course, voltage changes in neurons are not like step functions that we might use experimentally to measure the ion channel behavior. Instead, the voltage changes smoothly as the ionic currents charge and discharge the membrane capacitance. This leads to continuous changes in the currents through the channels. Here, we are referring to the macroscopic (time average) currents that we would record through the hundreds or thousands of channels in the membrane, not to the single-channel openings and closings, which are step-like. These smooth voltage changes reflect the voltage-dependence and voltage and time-dependent rates of channel opening and closing as the voltage fluctuates. Let's examine these.

Open the simulator, and select the "HH-AP". To make the traces easier to read, adjust the "Pulse" duration to 10 ms. Running the simulator with "Pulse Once" should yield a single action potential (use Capture Results). You should get a set of traces similar to those in Figure 18.

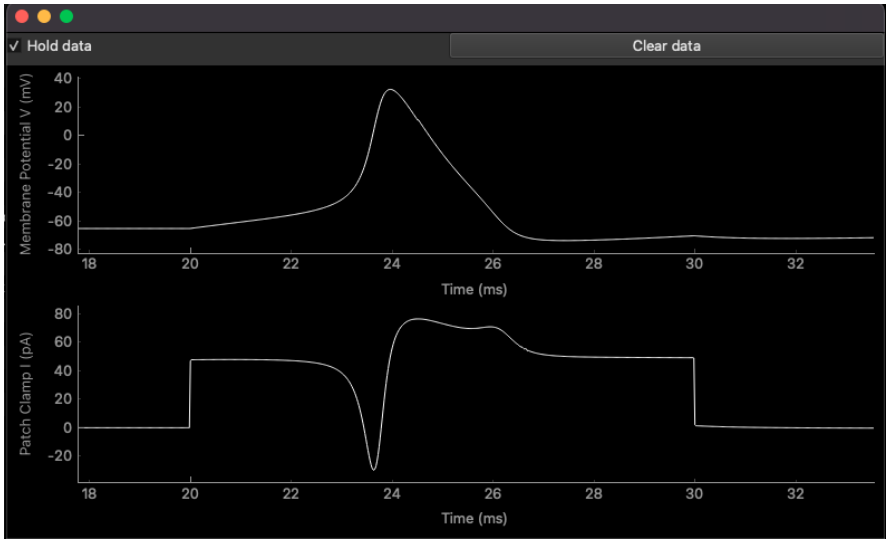


Figure 18

The upper trace is the action potential, while the lower trace shows the sum of the membrane currents. You can see that during the upstroke of the action potential, there is an inward (negative) current, which quickly decreases (goes positive), becoming an outward current, which then decays. Now, uncheck the "Plot Current" box, and in Ion Channels, check "Plot I" for both the soma.INa and soma.IK channels. Clear the analysis data, and pulse once again. This should give you a plot that looks like Figure 19, where you can see the separate currents.

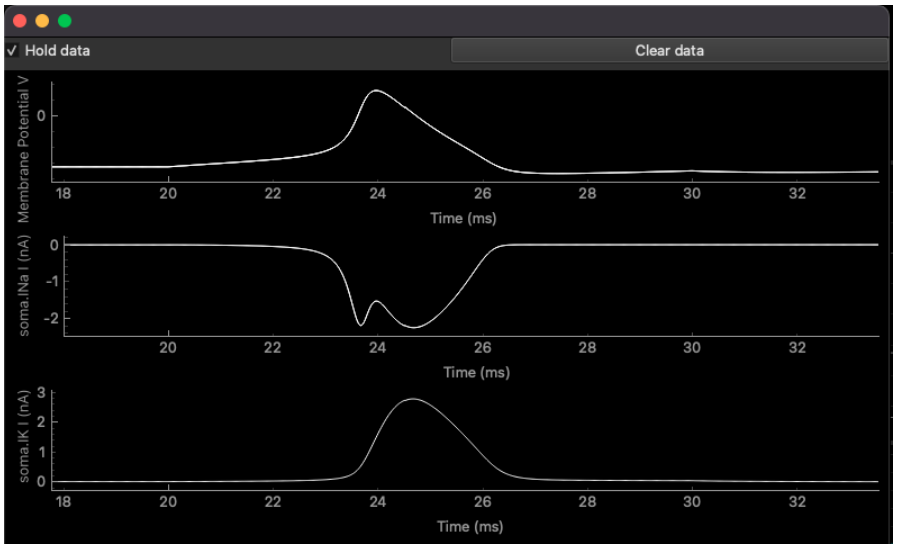


Figure 19 Sodium and potassium currents during an action potential.

Questions

a. The middle trace inf Figure 18 shows the sodium current. Why do you think this current has a funny shape with a "wiggle" in the middle? Can you think of a manipulation to test this hypothesis? Try it (remember, you can change some of the parameters in the Ion channel boxes).

b. The lower trace shows the current through the potassium channels. This has a nice smooth behavior. Compared to the action potential and the sodium current, when does this current start to activate ? When does it peak?
c. Check the Plot G (conductance) boxes for the sodium and potassium channels and "Run Once". Does the shape of the conductance plots help you understand the shape of the currents?
d. Switch to the "LG AP" preset for a different set of channel kinetics. Uncheck the HH IK and Ina plots, and check the INa2, IKf and IKs current plots, then "Run Once". Note that the action potential has a rather different shape. But the sodium and potassium currents have some similar features to those with the Hodkin-Huxley action potential. Look closely at these waveforms and see if you can explain each of their shapes.
e. Switch back to the HH channels. Do you think you can generate an action potential with just a potassium channel? Try it by setting the sodium conductance to 0 (or unchecking soma.Ina).
f. Do you think that the action potential requires a K channel to repolarize? Try it by reducing the K conductance, or even setting it to 0 nS. This is tricky to do, so first, run the simulator with the K channels at their normal conductance, then stop the simulator. Click the "Capture Results", and the "Pulse Once" button. What do you observe? Can you explain this result? Try setting the K conductance to values between 0 and the default 120 nS.
g. Can the leak substitute for the K channel? Turn off the K channel and increase the leak to 10, 20 or 50 nS. What do you observe? Can you explain these results?

Activation vs. Deactivation vs. Inactivation

The gating of Ion channels is often described as consisting of "activation", "deactivation" and "inactivation". In the K channel that we examined in Exercise 3, the channel only exhibited activation and deactivation. Activation represents the opening of the channel, usually with depolarization of the membrane. Once the channel has been opened, a change in the membrane potential to a more hyperpolarized level will cause some of the channels (perhaps all of them depending on the voltage) to close, which is deactivation. As noted above, the time course of activation is often sigmoidal, whereas the time course of deactivation is usually (but not always) dominated by a single exponential time course.

The third process is called "inactivation". Inactivation occurs when the channel closes, without a change the membrane potential. The inactivated state does not necessarily mean that the channel structure is in the same conformation as when the channel is in the closed state. Inactivation plays an important functional role in many channel types.

In Figure 19, it is difficult to discern when these processes are taking place because the membrane potential is continually and rapidly changing, which changes both the driving force for the ions and the opening and closing of channels. This is one reason that analysis using voltage-clamp is such a powerful tool, as it permits you to tease out the different aspects of the channel operation, by removing the changing voltage as an experimental variable.

Exercise 5: The Sodium Channel: Activation and Inactivation

As we have seen, the initiation of an action potential is driven by a positive feedback cycle where opening of the sodium channels depolarizes the cell, which then increases the open probability of more sodium channels, and so on, until the voltage reaches a peak value close to E_{Na}. At this point, the falling phase of the action potential begins, driven in part by the opening of the delayed rectifier potassium channels. However, that is not all: the sodium channels then begin to inactivate.

In the simulator, you can observe this in the HH-AP model. Set the Patch Clamp mode to VC, and turn off the K channel (leave the leak channels on). Make sure that all the values are reset to their defaults. Set the amplitude of the Pulse to 50 mV, and with Capture on, Pulse Once (and start the simulator if needed). To help things along, turn off Plot Current in the Patch Clamp panel, and check the Plot I box in the sodium channel widget, so that you are looking only at the sodium channel currents. What you will see looks something like this:

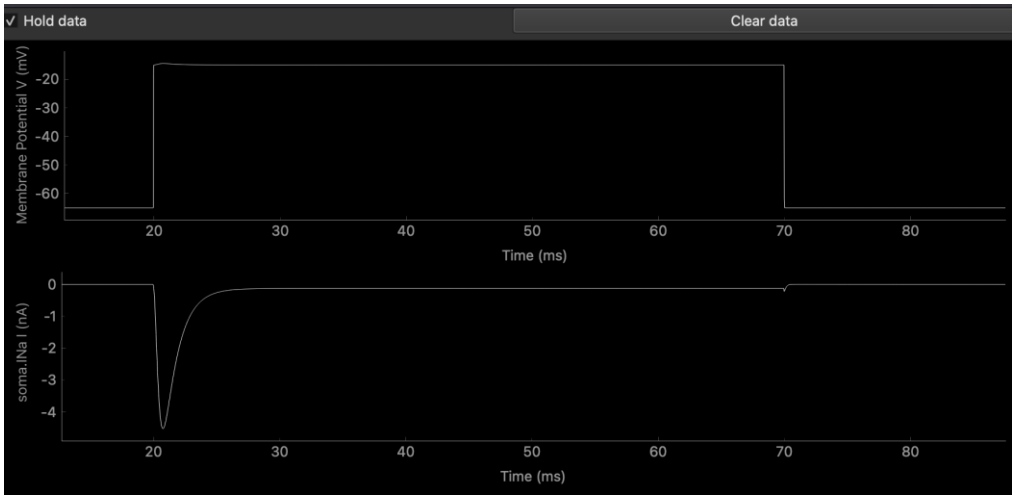


Figure 20

The upper trace is the command voltage, and the lower trace is the currents through the sodium channels. Note that the sodium channel current activates very rapidly, here in less than 1 ms. However then the current decreases almost as quickly to almost nothing. This decrease is inactivation, because the channels are closing even though the voltage is staying constant at -15 mV. There is a tiny steady-state current that remains, which is clear when you look at the end of the step.

Questions

a. Explain why the is current negative when the channels open.
b. Can you think of a way to determine whether the small (~150 pA) current during the later part of the pulse is from the sodium channels or not? There are at least 3 simple experiments that you can do to test you ideas.
c. Plot the current-voltage relationship of the peak current using the Pulse Sequence.

If we plot the sodium current over a wide range of voltages (steps from -40 to +150 relative to the holding of -65 mV), with 41 steps, then we get the plot in Figure 21. The measurement that is taken is the “mean” of the current over a narrow time range – this does not quite catch the peak current but it gets the shape right.

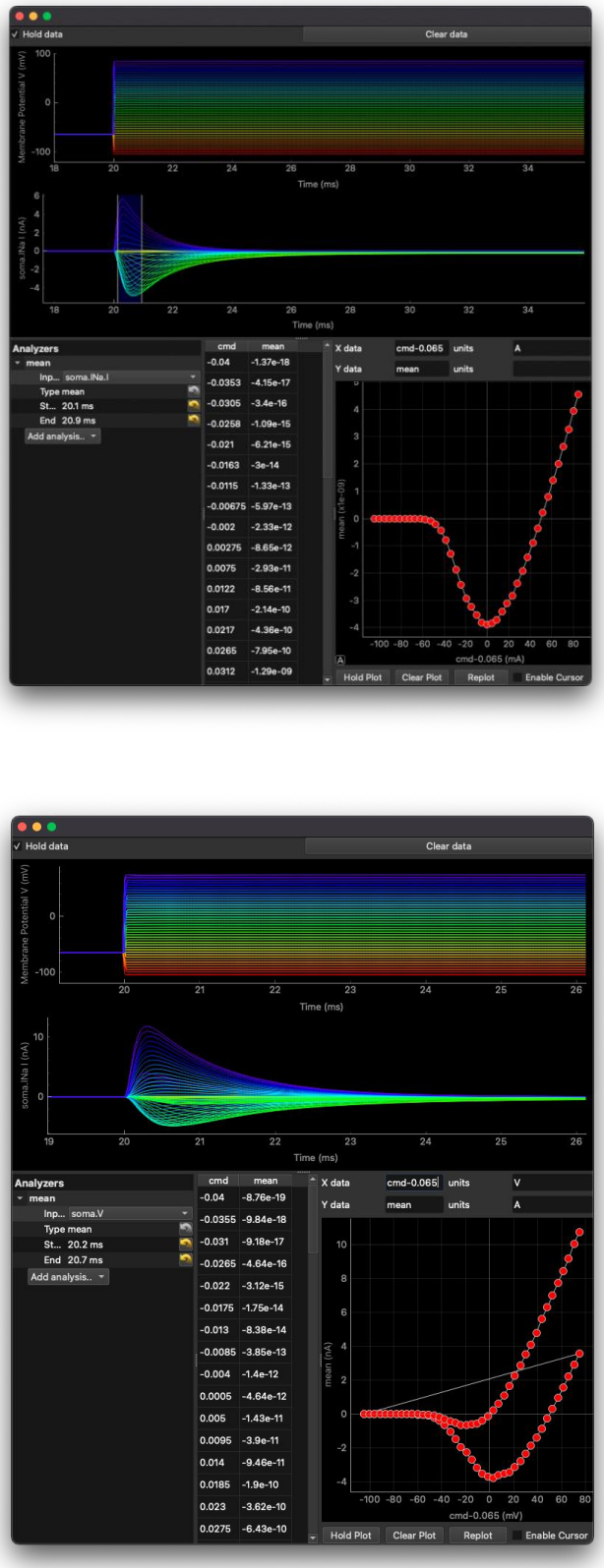


Figure 21

Why does the current-voltage plot in the lower right panel have the specific shape? What can you infer from this shape about the sodium channels?

Next, try varying the Erev for the Na channels (in the Ion Channels group). Setting the value to 0 and running the sequence again. The current-voltage plot now should appear as in Figure 22. Is the reversal potential for the currents what you expect?

Next, try changing the reversal potential to +200 mV. What does the curve look like now?

Try to calculate and plot the conductance. The conductance will be the mean current divided by the driving force (V-Vr), so the equation is:
Mean / ((cmd-0.065) – (0.0)) for the case where Erev was 0 mV. What happens, and why does this happen? It is easy to calculate the conductance when the voltages is far from Erev, but it is very sensitive to noise and other errors when close to Erev. In the simulations, we can get around this by plotting G for the selected ions, instead of I. Try it. Remember to change the Ydata to "mean" again for this plot, since G is already calculated.

Of course, you cannot directly measure the channel conductance, you must calculate it from the voltages and currents that you can measure. In a later section, we will discuss a different way of measuring the channel conductance that does not have the problem with the reversal potential that was seen above.

Gating variables

We can get more insight into how the sodium channels work by examining the two gating variables from the Hodgkin-Huxley model. For the sodium channel the activation variable is called "m", and the inactivation variable is called "h". The fraction of open channels is a product of these two variables, although we consider the cooperativity of the sodium channel as well. The resulting equation is:

$$g_{Na} = \bar{g}_{Na} m^3 h$$

This is similar to the equation for the potassium channel on page 23, but now includes the inactivation gating variable. To understand how the activation and inactivation work, first, lets plot m and h for the Na channel for a step to 0 mV.

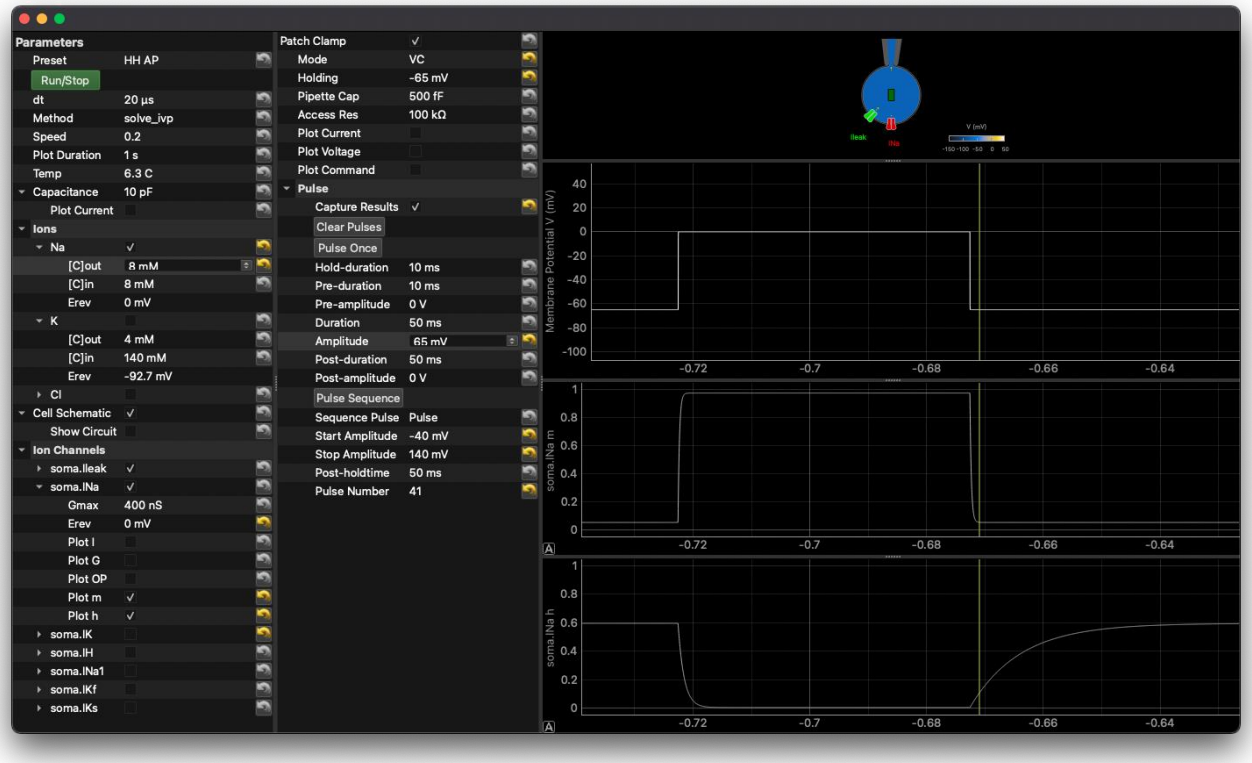
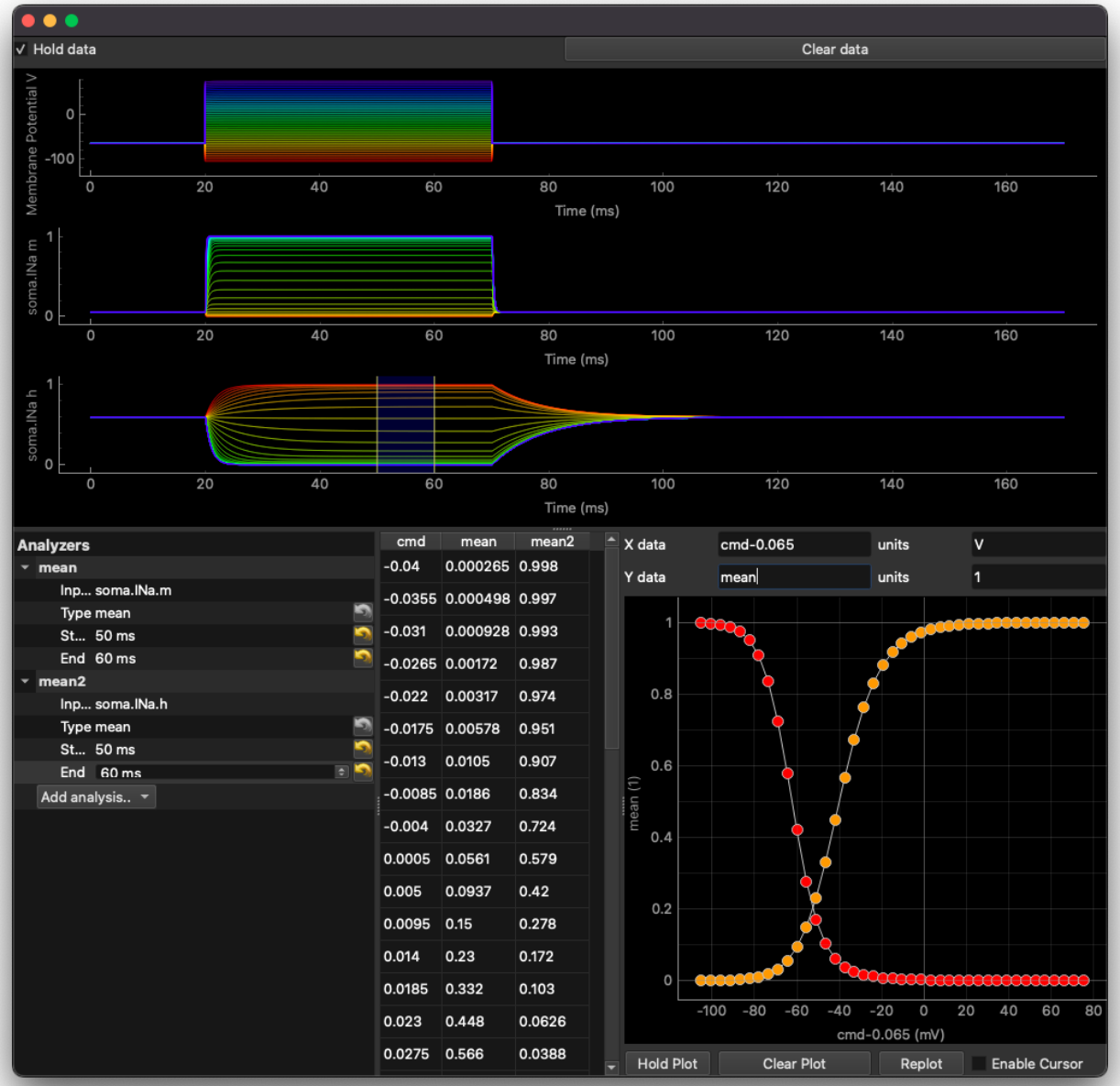


Figure 22 m (middle row) and h (lower row) for the sodium channel with a voltage step to 0 mV.

The first features you should notice is that m is small (about 0.08) at the holding potential of -65 mV, and rises quickly with depolarization to almost 1. However, h tells a different story. H is about 0.6 at -65 mV; this means that 65% of the channels are available to open, and about 40% are not available to open. During the step, the inactivation variable goes towards 0, but not quite as fast as the activation variable. As a result, the total fraction of channels available to open is large initially, but the channels rapidly inactivate. To further investigate how these variables change as a function of voltage, run a sequence of pulses (here, 41, from -40 to +140 mV relative to rest). Then in the analyzer, measure the mean currents for the two variables in a window from 50-60 ms (you may need to enter the times by hand rather than use the mouse). To plot both, first set the plot to x=cmd and y=mean, then click “Hold On’, and change the y to “mean2”. You should get two traces as in Figure 23.
H (in red) decreases with depolarization; that is fewer channels are available to open. M, on the other hand increases with depolarization. Note also that the inactivation gate changes more slowly with voltage than the activation gate. This slow behavior contributes to the relative and absolute refractory periods that we discussed above.

There are several other points about these curves that we will discuss in class. All channels (including K and Ca) that have inactivation have the same basic features as the inactivating sodium channels, so the principles that you observe here are widely applicable and important to understand neural excitability.

The last figure, just for fun, shows m and h during action potentials.


Hold data
Clear data
Membrane Potential V
soma.INa m
soma.INa h
Time (ms)
Analyzers
mean
Inp... soma.INa.m
Type mean
St... 50 ms
End 60 ms
mean2
Inp... soma.INa.h
Type mean
St... 50 ms
End 60 ms
Add analysis..
cmd mean mean2
-0.04 0.000265 0.998
-0.0355 0.000498 0.997
-0.031 0.000928 0.993
-0.0265 0.00172 0.987
-0.022 0.00317 0.974
-0.0175 0.00578 0.951
-0.013 0.0105 0.907
-0.0085 0.0186 0.834
-0.004 0.0327 0.724
0.0005 0.0561 0.579
0.005 0.0937 0.42
0.0095 0.15 0.278
0.014 0.23 0.172
0.0185 0.332 0.103
0.023 0.448 0.0626
0.0275 0.566 0.0388
X data cmd-0.065 units V
Y data mean units 1
mean (1)
cmd-0.065 (mV)
Hold Plot
Clear Plot
Replot
Enable Cursor

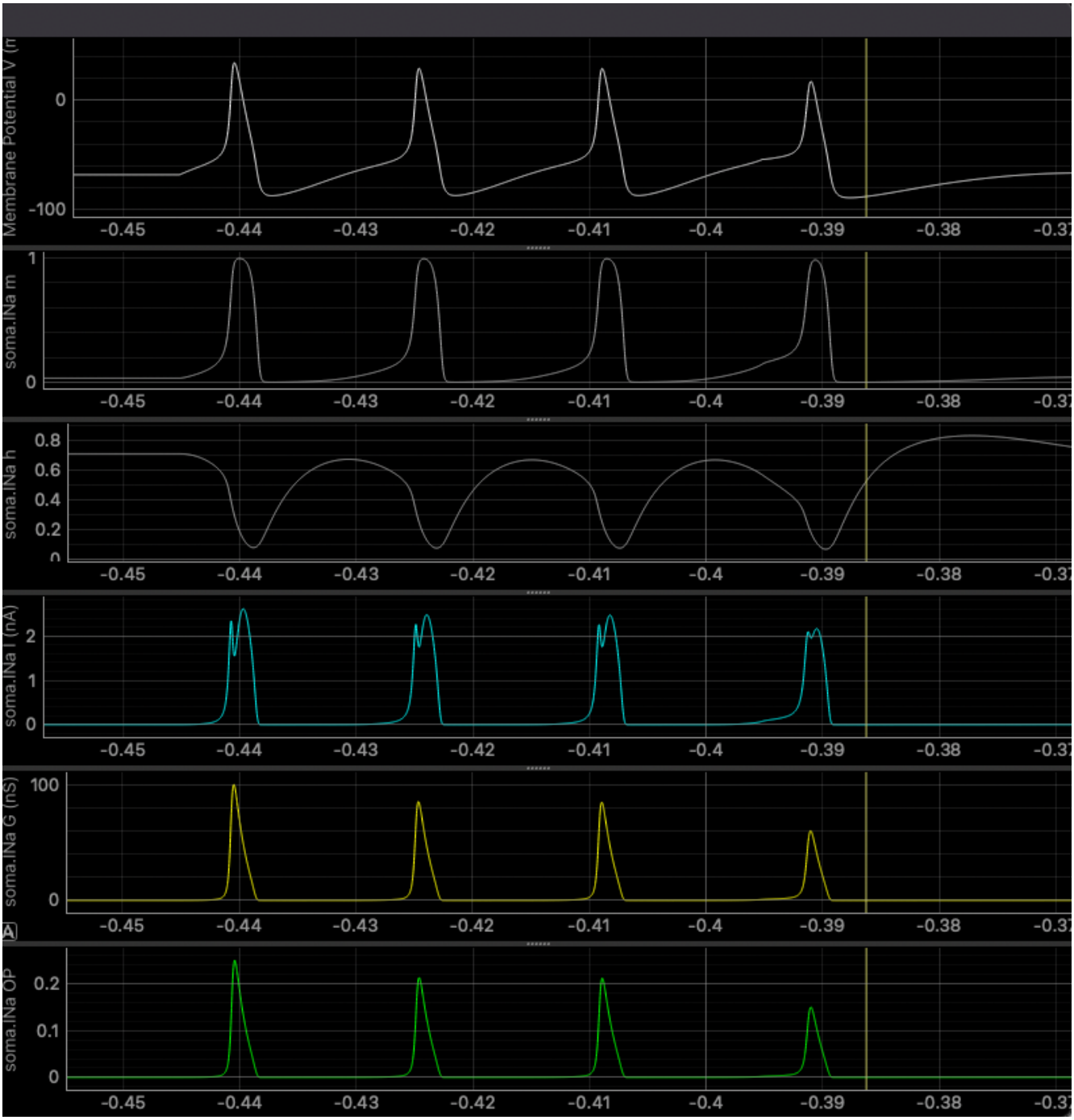


Figure 23 m, h, INa, GNa, and Popen for sodium channels during action potentials.

Exercise 6: Measuring “Reversal” potentials

Exercise 8: Measuring channel conductance with tail currents

References

Hodgkin, A.L. and Huxley, A.F. Currents carried by sodium and potassium through the membrane of the giant axon of *Loligo*. J. Physiol. 117: 544-472, 1952a.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392213/pdf/jphysiol01447-0082.pdf

Hodgkin, A.L. and Huxley, A.F. The components of membrane conductance in the giant axon of *Loligo*. J. Physiol. 117: 473-496, 1952b.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392209/pdf/jphysiol01447-0106.pdf

Hodgkin, A.L. and Huxley, A.F. A Quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117: 500-544, 1952c.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/pdf/jphysiol01442-0106.pdf

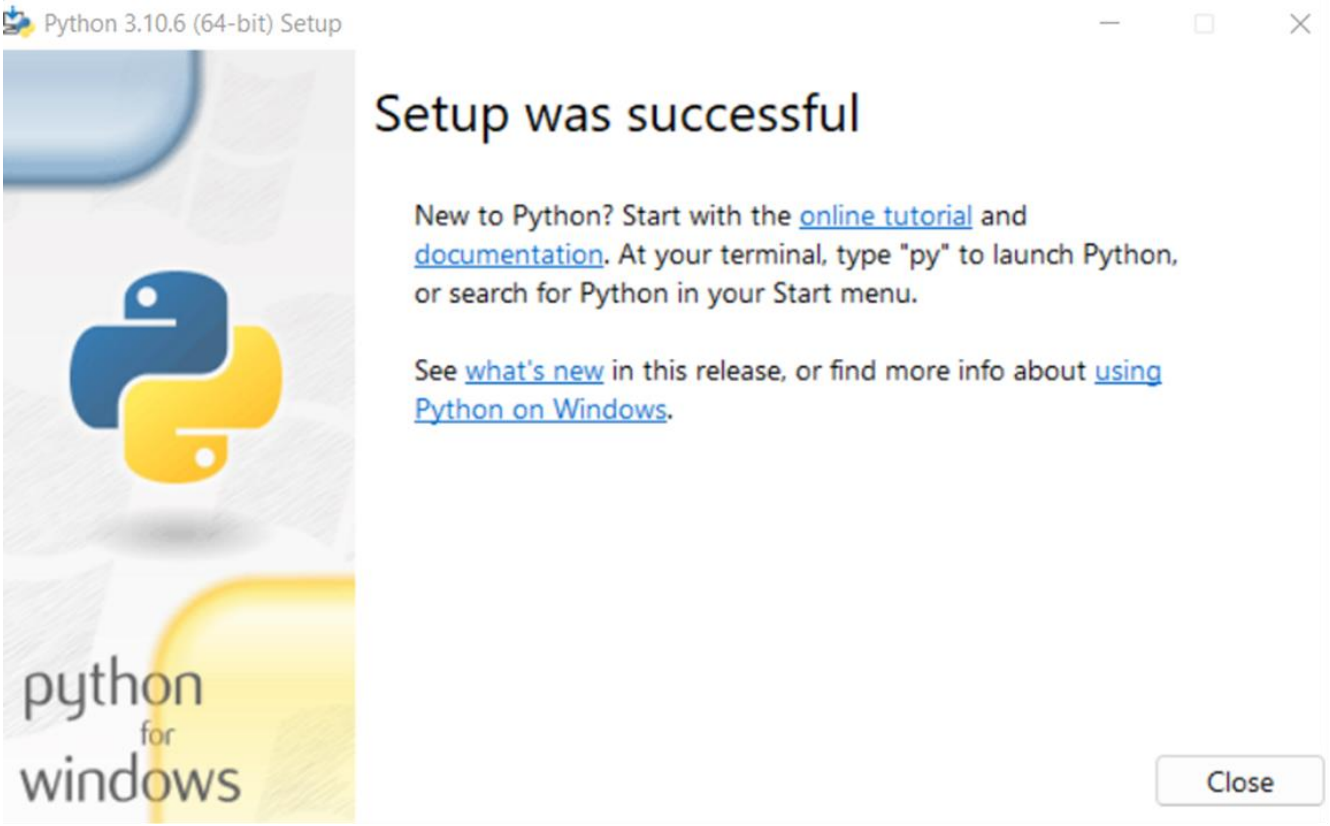
Appendix I: Installing Neurodemo for Windows from the git repository.

Windows:

Step 1: Installing Python on Windows and Adding it to your "Path"

- Go to: https://www.python.org/downloads/
- Click "Download Python 3.10.6" (or 3.10.7)
- Open the installer once it finishes downloading

- If you already have a version of Python installed you would see different options here. If you have issues running the demo install later in this guide, it is recommended to simply come back here then uninstall Python and start over with a fresh install and follow the steps here:
- *IMPORTANT: YOU MUST CHECK* "Add Python 3.10 to PATH" otherwise you will have to do this manually by editing system variables on your Windows computer later and that is a bit more complicated than letting the Python installer handle everything.
- Click "Install Now" after you have checked the box as shown in the picture above
- Once it finished the install you will something similar to this:



- If you see an option “override max bit value” in this screen it is recommended to go ahead and simply override the max allowed length. This should only take a few seconds at most to change. Then click "Close"
- Python is now installed and added to your computer's Path

Step 2: Download a copy of the code locally

- The easy way:
  - Go to: “GitHub: "https://github.com/pbmanis/neurodemo"” in your internet browser
  - Find the “Code” drop-down button near the top-middle of the page
  - Click “Download Zip”
  - Extract the .zip file – you can use the default location that it provides as long as you know how to get there in file explorer. Otherwise use another location that you decide.
  - Updating the code will require that you get the latest Zip file and re-extract it.

- Advanced way: An alternative is to use git at the command line to clone the repository with “git clone https://github.com/pbmanis/neurodemo”. If you do this you can update the repository as needed with “git pull” later on.

Step 3: Installing the demo

- Navigate to the folder where you extracted the .zip or cloned the repository.
- Find “win_install” and double click that “batch” file
  - Windows will have a pop-up warning of running programs it does not recognize,
  - Click “more info” and then “run anyway” will appear at the bottom of the pop-up
- A command prompt or command terminal will open and run some instructions
- This step will download several files that the neuro demo requires to run.
- Note: if you have updated the code, you *might* need to re-run this step. Try it first.

Step 4: Running the program

- You can simply double click “win_start_demo” and the program will run
- You can minimize the command terminal window that pops up along with the demo
    - FYI, if you exit the command terminal window it will stop running the code close the demo program
- Alternatively, you can right click the “win_start_demo” and create a shortcut of it on your desktop to simplify starting up the demo next use
    - Right click the file
    - Click “create shortcut”
    - Drag and drop the newly created “win_start_demo – shortcut “ file to your desktop
    - Double click the short cut and the demo will start up