

Electrophysiology and AxoClamp Tutorial

Created by Daniel Wagenaar for Bi/CNS 162

Caltech, April 2011

Introduction

Welcome to Bi/CNS 162! In the first half of this course we will focus on *electrophysiology*: the art and science of recording the electrical signals that living neurons use to communicate to one another. This is a nontrivial enterprise, because a typical mammalian neuron is about 20 micrometers in diameter, and its ability to drive electrical currents is correspondingly small: on the order of nanoamperes. Thus, if you want to record its electrical activity, you need a very sensitive recording instrument. (You obviously also need a very small electrode; we'll come to that later.) In this class, we will use the AxoClamp-2B amplifier by Axon Instruments. This tutorial will teach you the basics of how to use this amplifier to get clean recordings from neurons, that is, recordings that reflect the unperturbed activity of the cells most accurately.

A lightning-fast overview of a neuron from an electrical perspective

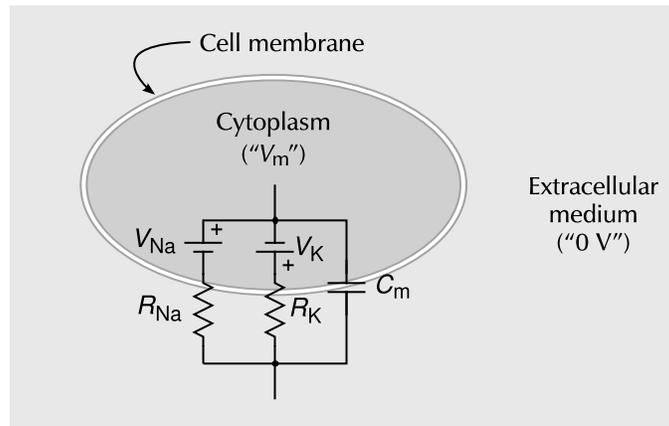
A neuron can be roughly modeled as a conductive sphere with an insulating membrane around it. The membrane is very thin and therefore has an appreciable capacitance¹, C_m , of about 10 pF. The membrane is also not a perfect insulator: it has some *leak conductance*, which gives it a membrane resistance, R_m , of about 50 M Ω . Finally, the membrane is not electrically passive: it contains ion pumps that maintain a difference between the concentration of various ions on the inside versus the outside of the cell. This concentration difference causes the resting potential of the cell's cytoplasm (its inside) to be negative relative to the extracellular medium, by about 70 mV in mammalian neurons.

There are other *voltage-gated ion channels* in the membrane that enable the cell to produce *action potentials*: brief depolarizations of the membrane potential that can propagate from the cell body along the *axon*, a long and thin appendage of the neuron, sometimes with multiple branches, that can project to nearby cells or all the way across a brain or even to distant parts of an animal's body. The end of the axon is marked by an *axon terminal*, where arriving action potentials lead to the release of chemicals known as neurotransmitters into the extracellular space. Typically, axon terminals are located in immediate proximity to other neurons' *dendrites*—shorter appendages of neurons that relay electrical signals to the cell body—and form *synapses* with those other neurons. At the location of the synapse, the *postsynaptic* neuron has a specialized structure known uncreatively as the *postsynaptic density* that contains ion channels that open specifically in response to the neurotransmitters released by the presynaptic neuron and that cause a small amount of current to flow into or out of the postsynaptic cell. This current ultimately leads to a depolarization or hyperpolarization of the cell body. If a neuron receives sufficiently many depolarizing inputs in a short enough time, its cell body may depolarize so much that an action potential is triggered. In this fashion, signals can propagate from cell to cell.

If the previous paragraph contained a lot of unfamiliar words, you should take a class like Bi 150 or Jerry Pine's Ph/Bi 103, or ask me for advice on a suitable textbook. For the purpose of today's class, however, we don't need to worry about axons and dendrites: we simply want to understand how recording from cell bodies works.

We can visualize the cell body, or *soma*, as follows:

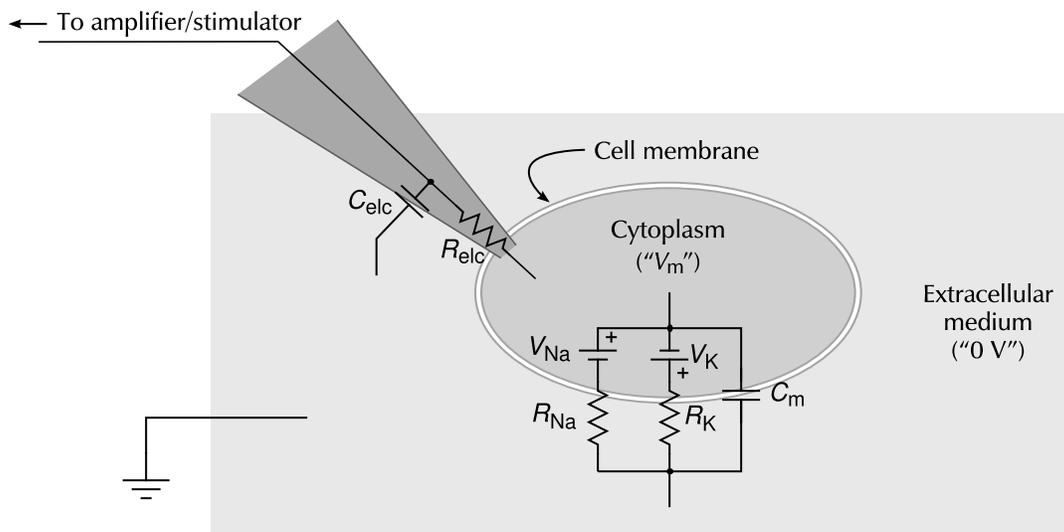
¹Don't quite remember what a capacitor or a resistor is? The "Recap of Basic Electronics" will refresh your memory.



Besides the membrane capacitance, the figure shows two batteries and corresponding resistors, for the two main ionic contributions to the membrane resting voltage. (Note that these are not meant to represent the (voltage-dependent) channels that give rise to action potentials; instead, these two resistors together are responsible for the major part of the leak conductance mentioned earlier.)

Connecting to neurons electrically

As electrophysiologists, we want to measure the membrane voltage (V_m) when the cell is at rest, as well as when we inject different amounts of current into the cell. This would be easy if we could just stick an alligator clip onto the cytoplasm, but of course the cell is much too small to contemplate such a thing. Instead, we have to resort to sticking a very tiny electrode into the cell. In practice, one uses a chlorided silver wire inside a glass micropipette filled with a strong electrolyte. This affords us a way to electrically contact the cytoplasm, albeit not quite directly:



The connection is indirect, because making an electrode small and sharp enough to penetrate a cell without doing major damage inevitably leads to a relatively large R_{elc} , typically at least $30\text{ M}\Omega$, that is, rather similar to the input resistance (R_m) of the cell. In addition, because the glass is so thin, there is an appreciable amount of stray capacitance between the inside of the micropipette and the bath (C_{elc}) which can be almost as large as the cell's capacitance (C_m).

In today's tutorial, you will gain an understanding of the properties of electrodes on intracellular recordings, and you will learn how to operate the AxoClamp-2B, a sophisticated piece of equipment that enables us to compensate for the effects of the electrode and obtain a nearly pure recording of the actual voltage inside a neuron. In order to decouple the challenges of learning to operate

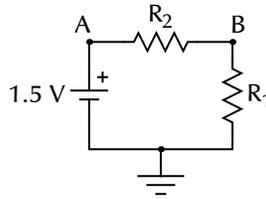
this machine from the challenges of sticking microelectrodes into living cells (and keeping those cells alive), we will construct model cells and model electrodes on a breadboard. Toward the end of today's class you will make your first attempts to record from actual cells, and we'll do a lot more of that over the next several weeks.

Getting to know the software

We're thoroughly modern in this lab, so we will use computer software to control most of our experiments and to record the results.

Recording voltages with *scope*

For those of you who grew up with oscilloscopes, it will be comforting to know that the recording software, *scope*, behaves very much like a virtual oscilloscope. For all others, I hope that the user interface will prove intuitive anyway. For our first test, we will build a simple circuit of a battery and two resistors in series:

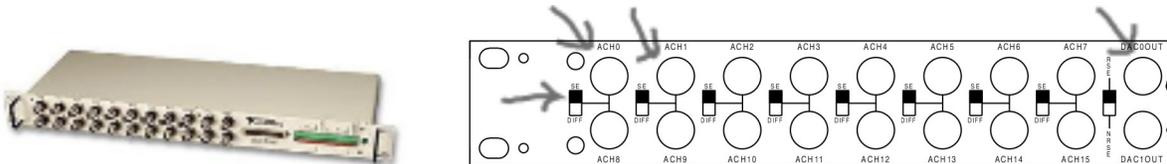


Suitable resistor values would be $R_1 \approx 2 \text{ M}\Omega$; $R_2 \approx 1 \text{ M}\Omega$.

As you may recall², the voltage at point *B* should be

$$V_B = V_A \frac{R_1}{R_1 + R_2}. \quad (1)$$

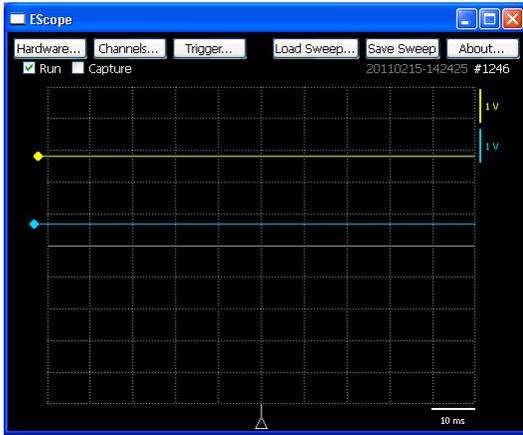
Build the circuit, and connect points *A* and *B* to channels 0 and 1 of your BNC-2090 data acquisition board. It is essential to connect the ground point too. (Why?)



The BNC-2090 data acquisition panel. The arrows in the schematic on the right point out several important features: The analog input channels (ACH) 0 and 1, the analog output channel DAC00OUT (which we will use later), and the “ground mode” switches, which all need to be in the “up” position.

Start *scope*, set it running, and check that relation (1) holds. You may find it instructive to pull out the connection to either or both points *A* and *B* and see the traces on the screen jump.

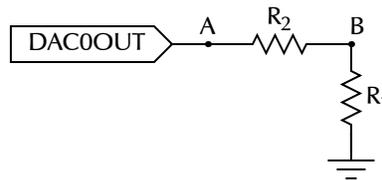
²If not, ask me to prove it to you based on the equations in the Electronics Recap.



The *escope* data acquisition program. The two lines represent to live data from two recording channels. The “ball” handles on the left indicate the zero-volt level for each channel. You can drag them around. To change the vertical scale of any channel, drag the scale bars (on the right). The time scale can be changed by dragging the bottom scale bar. To select which channels to display, click on the “Channels” button.

Stimulating with *espark*

To make the circuit slightly more interesting, let’s replace the battery by a variable voltage source. The BNC-2090 provides two convenient sources: DAC0OUT and DAC1OUT. Remove the battery from the circuit, and connect DAC0OUT to point *A*:



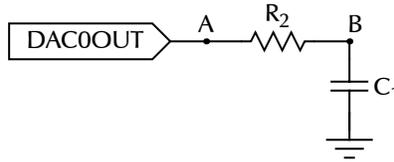
Start *espark*, and program in some pulses of your choice.



The *espark* pulse generation program. The program can generate synchronized pulse trains on two separate outputs, but for now, we only use one output (labeled “A”, corresponding, by default, to DAC0OUT). All aspects of the pulse (train) can be changed, and the little graphs give you an indication of what output to expect.

While *escope* is also running, click “Run” on *espark* and watch the *escope* screen to see if you can see the stimuli. (If they flew by too fast, set *escope* to “Trigger mode.”) Does relation (1) still hold?

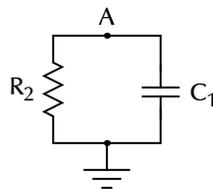
As a last non-biological example, let’s try a simple RC circuit:



Replace R_1 with a capacitor (10 nF would be a fine value), and apply the same pulses as before. You may remember that following a voltage step, a capacitor in an RC circuit like this slowly charges up. Do you see that happening here? The time it takes for the capacitor to charge up to $1 - \frac{1}{e} = 0.63$ times its final value is known as the “time constant” of the circuit, and should equal $R \times C$. Does it?

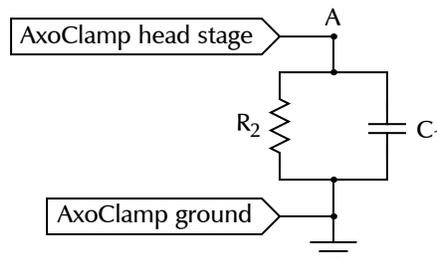
Injecting current with the AxoClamp

Before we start on building up our model of a neuron, we need to learn one more thing: how to use the AxoClamp as a current source. (We will want to inject known quantities of current into our model cells, and, as yet, we have only learned how to apply a known voltage, which is not the same. Do you see why?) Build the following RC circuit on your breadboard:



where R_2 is 1 M Ω and C_1 is 10 nF as before.

We will use the AxoClamp to inject current into this circuit. The AxoClamp has two “head stages” that can be used for this purpose, and they are not equivalent. In particular, one is capable of driving 10x more current than the other, which is quite useful for our purposes today. Find the head stage that is labeled “1x” and not “0.1x,” and connect its front end (the center hole within the white sleeve) to point A in the circuit. Also connect the back end of the head stage to ground:



Also hook up point A to an analog input (e.g., ACH0) so you can measure its voltage directly with *escope*. At all times, leave the “Mode” of the amplifier set to “Bridge”. “DCC” mode is not “DC Current mode.” I will be happy to talk to you about “DCC” mode later, or you can read about it in the AxoClamp user guide.)



The AxoClamp-2B. Several features relevant for current injection are indicated: the “DC Current Command” for electrode #1 (arrows on bottom right), the “Step Command” (circle on left), and the “I Display Select” (arrow on top). Note that there is also a “DC Current Command” dial for electrode #2. Check your wiring to figure out whether the head stage you are using is #1 or #2.

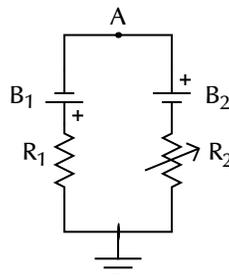
There are multiple ways to make the AxoClamp output a current. The easiest is to use the “DC Current Command” dial and switch. That allows you to directly dial in an approximate current. Try it. Noting that $V = I \times R$, use the voltage that you measure at point *A* to figure out what the numbers on the DC Current Command dial mean (or look it up in the Quick Guide).

Slightly more advanced is to use the “Step Command” manually, which allows you to inject a more precisely defined amount of current. Try it. Note that you need to set the “Step Command” destination before doing anything else. Again, use Ohm’s law to figure out what the numbers on the AxoClamp’s front panel mean. (Answer: for ME2, i.e., the 1x headstage, “+050.0” means a positive current of 50 nA.)

A much fancier way to inject current is to use *espark* to provide a timing signal to the AxoClamp. To do that, you connect DAC0OUT on the BNC-2090 to the connector labeled “Step Activate” on the back of the AxoClamp, and program *espark* to deliver 5 V pulses. Whenever the “Step Activate” input is driven high in this manner, the current selected on the front of the AxoClamp is delivered to the head stage. Try it. (The most fancy way to inject current is to have *espark* directly program the current waveform. To do that, you would connect DAC0OUT to either the “Ext ME1 Command” or “Ext ME2 Command” inputs on the back. But we don’t need to do that today.)

Modeling the cell membrane

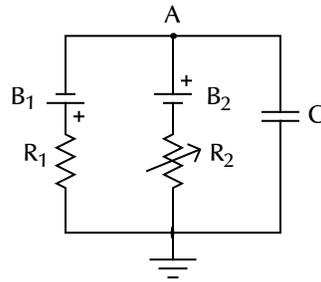
In the next several exercises, you will build up on your breadboard the model of the cell membrane introduced in the beginning of this tutorial. Let’s begin with the “ion channels,” represented by batteries and resistors, and worry about the membrane capacitance later. Build the following circuit on your breadboard:



where R_1 is about 50 k Ω and R_2 is a variable resistor or “potentiometer.”. Note that the two batteries are placed with opposite polarity.

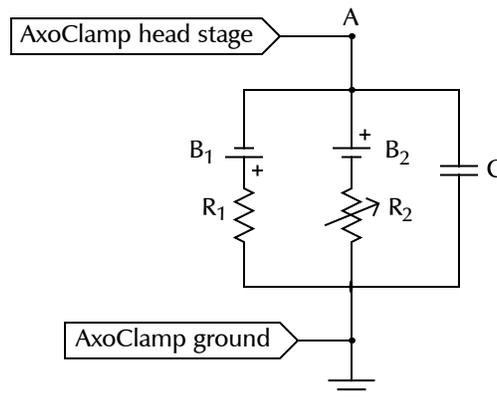
Point *A* represents the inside of the cell, so that V_A will be V_m . Can you predict, a priori, what value R_2 should take so that the voltage at point *A* is zero? Now, turn R_2 such that the voltage at point *A* is a realistic value for V_m .

To turn this into a realistic model of the cell membrane, all we need to do is to add the membrane capacitance back in:



where C is 10 nF.

For our first pseudobiological example, we will use the AxoClamp to inject current into this circuit, in a manner analogous to our previous RC circuit example:

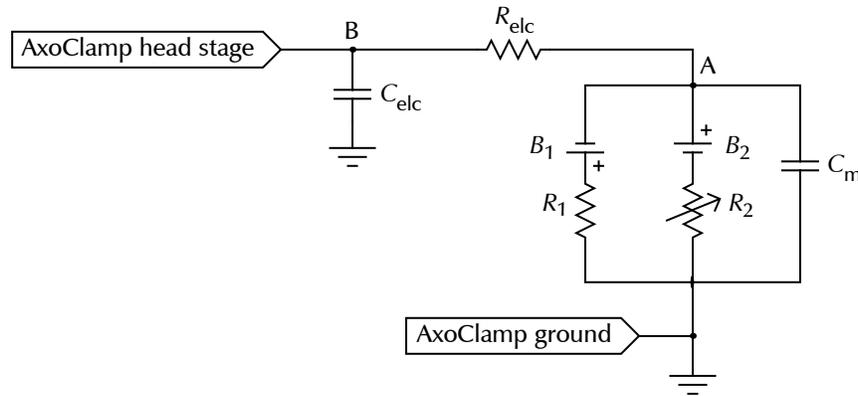


Try injecting a pulse of 50 to 100 nanoamperes into the circuit, measure the voltage at point A by connecting it to ACH0, and calculate the effective membrane resistance (again using Ohm's law). Does the value you calculated match the parallel combination of R_1 and R_2 ? At the onset and offset of the pulse, you can see the capacitor gradually charging up. Does the time constant make sense? (Remember, it should equal $R \times C$.)

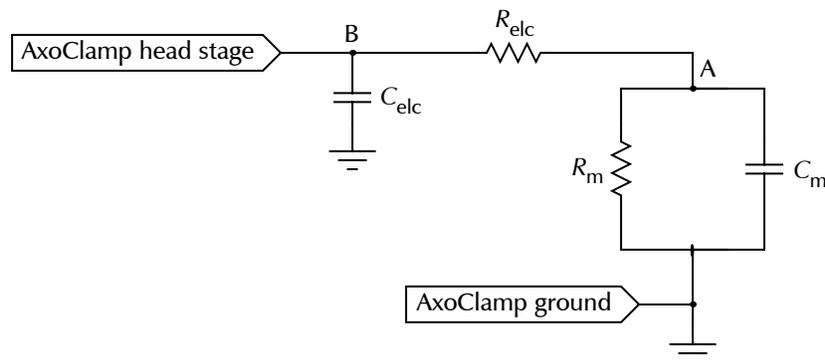
Modeling a cell and an electrode

If we could record from actual cells in the way we recorded from the circuit in the previous example, life would be good. Unfortunately, we cannot ignore the electrical properties of the microelectrodes needed to record from living cells. In the next exercise, we will study a more or less realistic model of an electrode stuck into a cell. Based on the two drawings on page 2, can you add a model of an electrode to the circuit diagram above? (Don't do it on the breadboard, just make a sketch in the space below. My solution is on the next page.)

Did you draw something like



? Building that whole circuit up on our little breadboard would involve quite a tangle of wires, so I propose we simplify it to:



This circuit behaves exactly like the previous one, except that the resting potential of the model cell is now 0 V.

Start with $1\text{ M}\Omega$ for R_m and R_{elc} , 10 nF for C_m , and 470 pF for C_{elc} . These are not realistic values, but they do result in more or less realistic time constants, and will allow for some interesting measurements. Connect points A and B to the analog inputs of your BNC-2090. Use the AxoClamp to inject 50 nA of current into the “electrode.” Is the voltage response of the cell (as measured at point A) affected by the presence of the complex electrode? (Consider both the steady-state result and the capacitive transient.)

It is worth repeating that in an actual biological experiment, we have no direct access to point A ; only to point B . If you compare, on the *scope* screen, the voltage trace measured at point B to the trace measured at point A , you will notice two differences: the most obvious difference is that point B experiences a larger steady-state voltage deflection than point A , as a result of the series resistance of the electrode (R_{elc}). A more subtle difference, which may be hard to see, is due to the fact that an electrode has a non-negligible capacitance between its inside and its outside. In our circuit, this is modeled by C_{elc} , and it affects the shape of the initial transient at point B . Fortunately, the AxoClamp contains circuitry to overcome both of these problems. As the final exercise before we advance to actual living cells next week, we will learn how to operate this circuitry.

Recording voltages with the AxoClamp

Thus far, we have only used the AxoClamp as an expensive current source. Now, we will put it to some real use. The AxoClamp contains two entirely independent modules for amplifying voltages from the two headstages. It also sports a dazzling array of knobs and dials on its front panel. I created a “Brief Guide to the Front Panel of the AxoClamp” to help you see through the thicket. Please read it before continuing.

Connect a cable between the appropriate voltage output of the amplifier (the BNC labeled V_2 on the top right of the front panel) and a third input of the BNC2090, and enable the corresponding trace in *escape*. When you inject a current (as before), does the V_2 trace match the recording from point A ? Point B ?

Compensating for the electrode’s properties

Probably, the answer was: neither. We will now use the three knobs “Output Offset,” “Bridge,” and “Capacitance Neutralization” to ameliorate the situation.

First off: Real electrodes feature an electrode–electrolyte interface that can act like a small battery. Therefore, reading absolute voltage values is not trivial: any measurement you take may be corrupted by a (hopefully constant) offset. The “Output Offset” button allows you to compensate for it. Try turning it left or right until (in the absence of stimuli) the voltage from V_2 matches the voltage at point A and B , which should be zero. Make sure the “Capacitance neutralization” and “Bridge” knobs of the AxoClamp are all the way off (turned counter-clockwise).

Use either the “DC Current Command” or the “Step Command” to inject a current pulse into your model. If all is well, the output from V_2 should match exactly the direct measurement from point B . Check that it does.

What we want, is for the output of the AxoClamp (i.e., V_2) to match the voltage at point A instead. There are two knobs on the AxoClamp to approximate that goal. The first is the “Bridge” dial. By turning that up, you can compensate for the electrode’s series resistance. Try it. Can you set the dial such that the steady-state output of the AxoClamp matches the steady-state voltage at point A ? Does the number of the dial match the actual resistor value that you used for R_{elc} ?

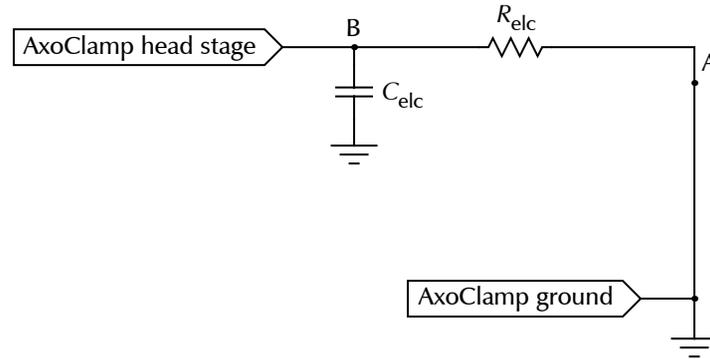
If you successfully got the steady-state output to match, you can probably see the transient due to the electrode capacitance more clearly. The AxoClamp contains specialized circuitry to get rid of that. To get rid of it, try turning up the “Capacitance Neutralization” knob.

Probably, turning the “Capacitance Neutralization” knob didn’t have much of an effect. That is because the value we used for C_{elc} is unrealistically large. To see the effect of capacitance neutralization, we need to use more realistic component values. Reasonable values for the components are: R_{elc} : 30 M Ω ; C_{elc} : 10 pF; R_{m} : 30 M Ω , and C_{m} : 470 nF. Unfortunately, with such large resistors, we will not be able to use the BNC-2090 for a direct measurement from points A or B . (Its internal circuitry would too strongly influence the delicate circuit on your breadboard.)

So, we will only use the AxoClamp. Unfortunately, that means that we will not be able to directly compare the “real” voltage at points A and B with the measured value for V_2 . How then can we know that we are setting the knobs to the appropriate position?

A realistic model electrode

Let's start out with *only* a model electrode on our bread board, and no cell, because then we know what we want to see at V_2 : absolutely nothing, even when we stimulate. So build the following circuit:

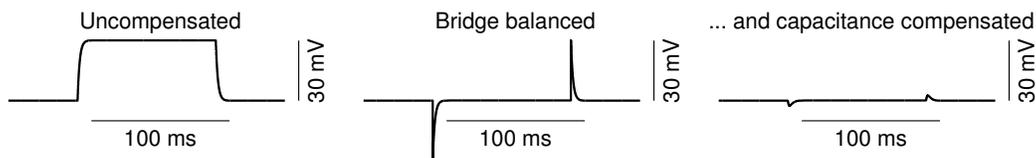


where R_{elc} is $30 \text{ M}\Omega$ and C_{elc} is 10 pF . This corresponds to the situation when a real electrode is dipped into saline, but not stuck into a cell.

First, turn the “Bridge” and “Capacitance Neutralization” knobs all the way off. Then, tune the “Output Offset” until the LED display and *scope* indicate zero volts for V_2 in the absence of stimuli. Now, get the stimuli going again. Can you balance the bridge? Try doing that by looking only at the *scope* screen. Once you are satisfied, look at the “Bridge” knob, and see if the numbers make sense. (They should match R_{elc} .)

Next, use *scope* to zoom in on the electrode transient, and gradually turn up the knob. Do you see the transient getting shorter in duration? Can you tune it all the way away? If not, turn the knob all the way back down, and estimate, from the screen, the time constant of the transient. Remembering that $\tau = RC$, does the time constant make sense? Probably, your time constant is larger than what you would expect from the component values in your circuit. That is because there is a fair amount of “stray capacitance” in circuits like this. If you pull the C_{elc} capacitor out altogether, you should see the time constant get shorter (by $\Delta\tau = R_{\text{elc}} C_{\text{elc}}$), but not go away. What remains is due to the stray capacitance, which is typically around 10 pF . (Calculate the stray capacitance in your circuit, and see if it similar.)

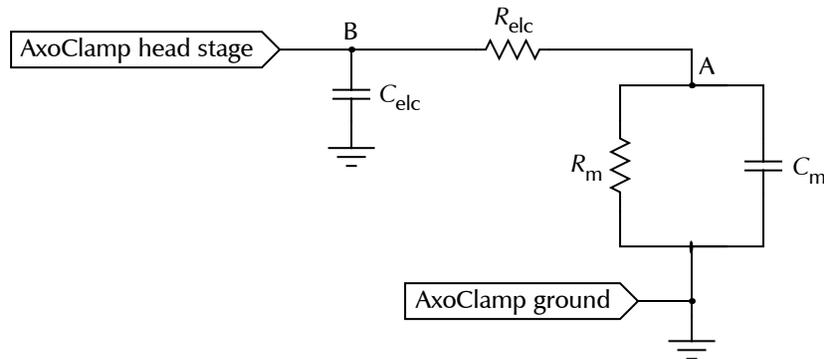
If you couldn't compensate for the full capacitance earlier, try turning up the “Capacitance Neutralization” now, to see if you can compensate for the stray capacitance. You probably can. What happens when you overcompensate? This “ringing” can easily kill a cell if you make it happen when your electrode is inside one, but you can also use it to your advantage: briefly ringing often helps an electrode get into a cell. In fact, that is just what the “Buzz” button does: it temporarily increases the capacitance neutralization beyond its stable range.



Electrode traces in response to a 100-ms long current pulse of 1 nA , when the electrode is not in any cell.

A realistic model cell

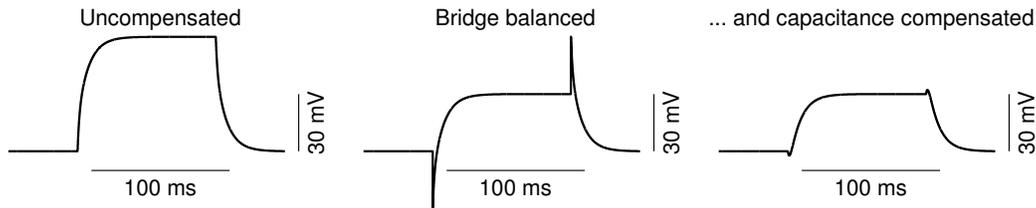
Finally, let's return to our full electrode-plus-cell model:



but be sure to turn the “Bridge” and “Capacitance Neutralization” knobs all the way down before you change anything on the breadboard. Use the component values suggested earlier, and if you previously had to pull C_{elc} out to achieve capacitance neutralization, leave it out. The same stray capacitance will still be there.

Can you balance the bridge? How do you know which part of the signal is due to the electrode and which is due to the cell? This can sometimes be a bit of an art, but in favorable circumstances, the time constant of the electrode is much shorter than the time constant of the cell. You can use that to your advantage, by turning the “Bridge” balance up until the fast transient is just barely compensated for, but the slow part is still left in place. As before, see if the numbers make sense.

Can you compensate for the capacitance?



Electrode traces when the electrode is inside a neuron.

Conclusion

Congratulations! You have successfully recorded from a realistic model neuron. You are now ready to do the real thing, and next week, we'll have ganglia ready for you.