Pascal Nieters
Exercise sheet 1
May 5, 2023

Neurodynamics 2023

Homework: The Hodgkin-Huxley Model.

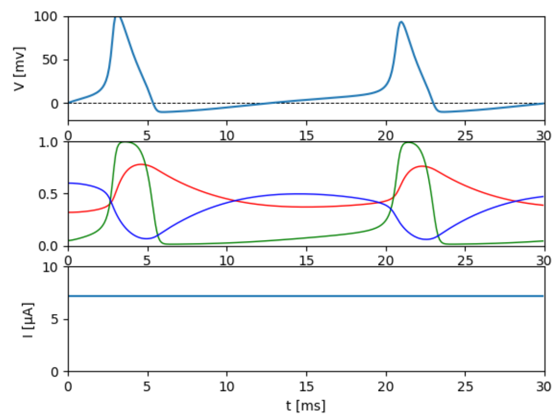
Your first job is to investigate the Hodgkin-Huxley model in detail using computational tools. We recommend using `brian2`[1], a python package for simulating neural networks. `brian2` will give you easy access to numerical solving algorithms for differential equations, our modelling language of choice. While the package comes with "neural speak overhead" when formulating problems, its wide use in the theoretical/computational neuroscience community means your investment in learning it will result in a strong basis for future work in the field – should you be interested.
Therefore, your unofficial first task is to get `brian2` up and running. Our recommendation is to set up a virtual python environment that also includes `jupyter`[2], `numpy`[3], `scipy`[4], and `matplotlib`[5]. The practice session on Friday, the 14th, and the tutorials are opportunities to get help getting started.
Once you are up and running, you will:

Question 1. Implement the Hodgkin-Huxley model as a `brian2` differential equation that you can simulate for a period of time and with different input currents I. It will be helpful to write a function `simluate_hhx(time_span, input_current)` that does this for you in a single call. This makes computational experiments much easier. `brian2` also provides `TimedArray`s which you can use to describe the input current, and `brian2.StateMonitor` that you can use to collect measurements of the state variables of the dynamical system.

Question 2. Using your implementation, reproduce the plot below using the following parameters:

C	1 µF
g_{K}	36 mS
g_{Na}	120 mS
g_{L}	0.3 mS
E_{K}	-12.0 mV
E_{Na}	120 mV
E_{L}	10.6 mV

$\alpha_n(V)$	$0.01\frac{10-V}{\exp\frac{10-V}{10}-1}$
$\beta_n(V)$	$0.125\exp\frac{-V}{80}$
$\alpha_m(V)$	$0.1\frac{25-V}{\exp\frac{25-V}{10}-1}$
$\beta_m(V)$	$4\exp\frac{-V}{18}$
$\alpha_h(V)$	$0.07\exp\frac{-V}{20}$
$\beta_h(V)$	$\frac{1}{\exp\frac{30-V}{10}+1}$



[1] https://brian2.readthedocs.io/en/stable/
[2] https://jupyter.org/install
[3] https://numpy.org/install/
[4] https://scipy.org/install/
[5] https://matplotlib.org/stable/tutorials/introductory/index.html

Question 3. Apply a step current: Turn on the current after 10ms to a value c for 40ms total, then turn it off. You can use the `input_current` function for this. What is the lowest current you can apply so that the neuron emits at least one spike? What is the lowest current you can apply so that the neuron fires repetitively? Using your plots of the simulation, characterize the difference between the repetitive and non-repetitive firing regimes.
In each experiment, simulate for at least 20ms after current offset.

Question 4. Apply a ramp current: Turn on a current that linearly increases to 12µA starting at $t_0 = 10$ms and lasting for t_{duration}ms until $I(t_0 + t_{\text{duration}}) = 12$µA. Find the shortest t_{duration} such that the neuron does not spike by starting at $t_{\text{duration}} = 60$ms and lowering the duration. Record the Voltage at $t_0 + t_{\text{long}}$.
Next, increase the current to only 4.5µA and find the shortest duration t_{short} such that the neuron does spike in the analogous way.
Plot not only the voltage, but also the three gating variables m, n, p and link the difference in the spiking threshold to their dynamics

Project Proposal: Investigate axonal propagation delay.

In essence, the Hodgkin-Huxley model describes the dynamics of any patch of membrane containing sodium and potassium channels. It was developed on a patch of the squid's giant axon. Extending the model to a partial differential equation that describes not only how the membrane voltage changes in time but also in space along the axon, we can investigate how neural action potentials travel along the axon where they can trigger synaptic transmission.
Investigate and then explain to your fellow students how quickly, or slowly, action potentials travel along a model axon with or without myelination.