2D1431 Mathematical Modelling of Biological Systems

Exercise 3 II Neuron and synapse models

1 Objectives

In this exercise you will get acquainted with some of the concepts in biologically realistic modeling of neurons and neural networks. When you are finished you should:

- have some hands on experience in running neuron simulations using the Swim simulator
- understand *passive membrane properties*
- understand how *action potentials* are generated
- understand the concept of *spike train adaptation* and the mechanism which generates it.
- understand the concept of *temporal* and *spatial summation*

2 Tasks

You will use the Swim simulator to experiment with the properties of model neurons, changing their membrane properties, structure, ion channels and connections to other neurons. Most of the tasks consist of modifying simulation specification files and observing the resulting changes. Examination consists of the examinator asking you about your observations.

The first part of this exercise will deal with passive membrane properties: how does the cell act as an electrical system? After that, voltage-dependent ion channels will be added to the model, and the action potentials will be observed. As more channels are added, you will see how the spiking frequency and spike shape can be changed. Finally, you will experiment with a model of synaptic connections, and see how postsynaptic potentials are integrated.

3 Background

This exercise introduces computer modeling of neurons and neural networks on a fairly detailed and biologically realistic level. Concepts like passive membrane properties, voltage dependent ion channels, and synaptic transmission are introduced and exemplified. These instructions are intended to be used together with the Swim simulator and a set of sample files prepared for this purpose.

Here, we concentrate on the single cell and synaptic interaction between cells. The simulations are based on a multicompartment model: the neurons are modelled as spherical or cylindrical compartments representing the soma and dendrites. Each compartment is assumed to be isopotential, electrically linked to neighboring compartments. Ion channel populations are modelled as variable conductances. The whole system can be modelled with an equivalent electrical circuit. This model has been used to simulate individual neurons and different biological neural networks.

All simulations are done using the *Swim* simulator[1] developed at the department of Numerical Analysis and Computing Science at the Royal Institute of Technology in Stockholm. More information about this simulator can be found in [2] and [1].

4 Using the Swim simulator

Copy the files of this exercise from `/info/biomod02/lab3` to your lab directory:

```
> cp -pr /info/biomod02/lab3 .
> cd ~/lab3/neuronmodellerII
```

These files are also listed in the appendices.

The Swim neural network simulator is actually a set of programs running in a Unix environment. Here, we will only use two of the programs: `swim` and `xswimgraph`. `swim` is a combined compiler and numerical simulator. It reads special *specification files*, translates them to an internal representation and performs the actual numerical simulation. `xswimgraph` is a program for plotting the output from `swim` in a window on an X terminal. It can also be used to send these plots to a PostScript printer.

A typical simulation is run by typing a command line like this:

```
> swim spec-file | xswimgraph -direct -timeline -labels
```

(here, a couple of optional switches have been used to make the resulting diagrams easier to interpret). To make it more convenient for you, we have prepared a shell script (command file) called `runswim` which runs the two programs as above. Thus, while working with this exercise you can run simulations by typing:

```
> runswim spec-file
```

When you run a simulation like this, `xswimgraph` will display a window on the screen with the selected state variables plotted as a function of time. For every simulation you run, you will get a new window on the screen. These windows can be removed by pressing the 'Q' key (for "quit") while the cursor is in that window. If you have a PostScript printer as the default printer, you can also get a printed copy of the plot by simply pressing the 'P' key (do **not** press it many times; you will print one copy for every 'P' typed!).

The specification files are written in a specially designed specification language. This language is described in detail in [2]. You will only have to modify some parameters to do experiments with them.

There are some details about the syntax that you should be aware of:

- Parameters usually have multiple word names (like "`Na G`", "`Stim Start`", etc).
- Parameters are assigned values on separate lines.
- Line breaks and indentation is significant (*i.e.* don't change it).

[1] This edition of the exercise is compatible with version 1.5 of the Swim simulator.

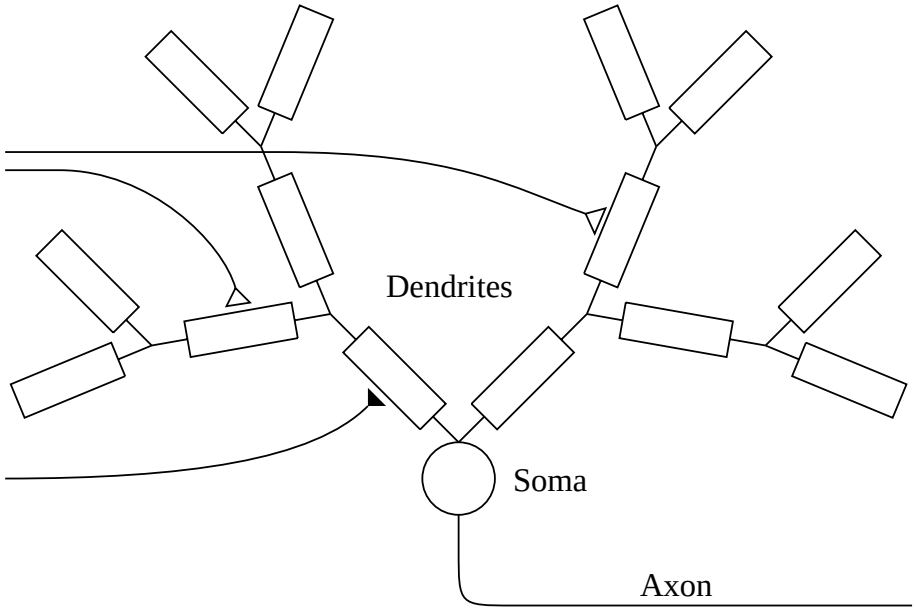

Figure 1: In this exercise, we will mainly use a 15 compartment model of the cell. One compartment represents the cell body (soma) and the others represent various parts of the dendritic tree.

5 Passive cell properties

The model neuron is built around a set of electrically coupled isopotential compartments. You will be using both a single compartment version and one where the compartments are arranged in a tree-like fashion (figure 1) to resemble the structure of a real dendritic tree.

The primary state variable of each compartment is the membrane potential E. This is the electrical potential on the inside of the membrane as compared to the outside. This potential is influenced by currents caused by moving ions.

A cell contains several so called ion pumps maintaining different concentrations of different ions on the inside and the outside. Since the membrane is not equally permeable to all ions, these concentration differences give rise to a potential gradient. For a resting neuron the potential on the inside can be around $-70\,\mathrm{mV}$. One usually refers to this effect as the *polarization* of the cell. Currents causing the potential to become even more negative are called *hyperpolarizing* while currents raising the potential are called *depolarizing*.

Electrically, each compartment has a capacitance and a conductance across the membrane and they are coupled by other conductances. One important property of the passive circuit is the time constant of the membrane. It controls *e.g.* the time it takes for the cell to reach equilibrium when the stimulation changes. The timeconstant is simply the ratio between the capacitance and the conductance across the membrane.

5.1 Single compartment

We will start by simulating a single isopotential compartment. We are looking at the membrane potential E as we inject current into the compartment.

Look at the file `passive` using the text editor. This file describes one neuron (called "N") receiving a stimulating current, but the current value is initially set to

0 nA. The current is injected into the "soma" compartment (which happens to be the only compartment).

- Run a simulation by typing: `runswim passive`
- Inject some current into the cell by changing the value of `Stim I` to 0.2e-9 (*i.e.* 0.2 nA). Rerun the simulation. What happens?

Note that it takes some time to depolarize the cell. This is because the membrane acts as a capacitance that has to be charged.

5.2 Dendritic tree

Now we will replace the single compartment with a multicompartment representation of the dendritic tree. We will use a tree structure with 15 compartments as shown in figure 1.

- Change the word `Passive` to `Passive_Tree` in the neuron header. You are now using a different predefined dendritic model from the "`default-cell.spec`"-file (that file is listed in appendix B, but you don't have to bother about the details now).
- Run a simulation. You will notice that the change in potential is much smaller (why?).
- Change the current to a larger value to get about the same size of the depolarization.

You have been injecting current into the soma part of the cell. In real cells it is very hard to inject current into the dendrites, but we can do that in the model to get an idea of what happens when synapses give input far out on the dendrites.

The compartments are labeled by a numbering scheme as shown in figure 5.2. By replacing `<soma>` with other labels, current can be injected into any compartment.

To make it simpler to compare the results of your changes it is convenient to simulate two cells with almost identical parameters, except for the most recent changes. You can create two neurons by simply duplicating the text starting with `neuron` and ending with `end`. Make sure that you also change the name "N" of one of the neurons, since two neurons are not allowed to have the same name.

- Make two identical neurons with different names.
- Give a short current pulse by setting `Stim Start` to 0.100 and `Stim Stop` to 0.105. This corresponds to a 5 ms pulse after 100 ms. Do this for both neurons. Simulate this.
- Replace `<soma>` with `<1>` for one of the neurons to inject the current into a compartment close to the soma. Simulate.
- Move the current injection to other compartments. You might also have to change the current strength and duration.

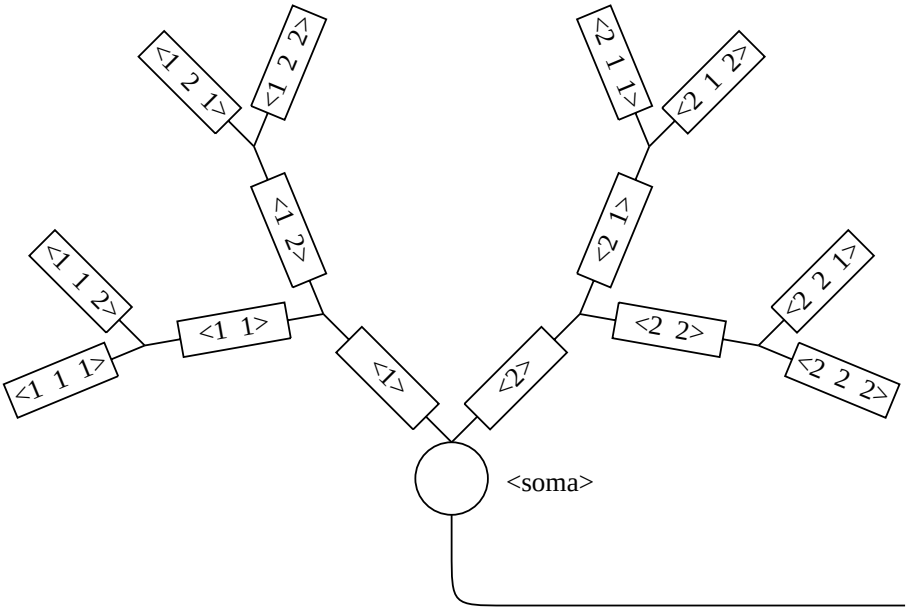

Figure 2: This is how the dendritic compartments are labeled. Note that the digits are separated by spaces (it is a common mistake to write <11> instead of <1 1>).

6 Spike generation

The most striking aspect about neurons is that they can generate *action potentials* (*spikes*). The action potential is caused by ion currents through selective sodium (Na^+) and potassium (K^+) channels, primarily in the soma and in the axon. These channels will allow sodium and potassium ions respectively to penetrate through the membrane. These channels are not passive in the sense that they can be described by an equivalent passive electric circuit. Rather, they open and close stochastically with probabilities depending on the membrane potential.

Each neuron contains thousands of such channels so we will not even think about trying to simulate them individually. From our point of view, the important aspect of the channels is the total electric current flowing through them. In 1952, Hodgkin and Huxley published their classical paper where they presented a model describing the collective effects of these ion channels. Basically, each type of channel is described by a state variable representing the probability of such a channel being open. This variable depends on the membrane potential and a time constant which is also potential dependent. The fraction of the channels being open is equal to this probability of each channel being open. Thus, the electrical conductivity is proportional to this probability.

We will put active channels only on the soma compartment while the dendritic compartments are basically passive. The dendrites of real neurons are not really passive but the active effects in the soma and axon usually dominate. The propagation of the action potential along the axon is not modeled in detail; only as a constant time delay. This is the reason why we do not need any axon compartments.

The action potential is initiated by the opening of the sodium channels that takes place when the cell is depolarized (*i.e.* the inside potential is raised). Opening of these channels cause sodium ions to flow into the cell making it even more depolarized. If nothing else would happen, the membrane potential would eventually stabilize at the equilibrium potential for the sodium ions at around +50 mV.

There are, however, two other things happening shortly afterwards. Firstly, the

sodium channels tend to enter an *inactivated* state after some time of depolarization. This is another closed state, so the potential will not stay depolarized for very long.

Secondly, potassium channels are also activated by the depolarization along with the sodium channels. They are slower than the sodium channels, but after a while they catch up. The equilibrium potential for the potassium ions is around $-80\,\mathrm{mV}$ because these ions primarily flows out from the cell. Thus, when the potassium channels open, shortly after the sodium channels, the membrane potential goes down again, often even slightly below the resting level at $-70\,\mathrm{mV}$.

6.1 Sodium channels

We will start by adding sodium channels to the soma compartment of our cell. We will use predefined values for all the parameters needed to specify the dynamics of this channel.

Look at the file `spikes` which specifies two neurons called "N1" and "N2". "N1" is identical to the one you have just used while "N2" has sodium channels added to the soma compartment.

- Simulate. Notice that you get a rapidly depolarizing spike followed by a slow repolarization. Why does the potential go down again? Why do you get only one spike?
- Run several simulations while varying the injected current. Try to determine the threshold when the cell fires.

6.2 Potassium channels

Now we will add potassium channels to the soma to be able to have complete spikes.

- Add both sodium and potassium channels to "N1" by inserting "Na K" in the list of included properties. (Look at how "Na" properties are included for "N2").
- Look at the shape of the spikes. How is it different from sodium only spikes?
- Now, you should be able to get repetitive firing fairly easy. Vary the current to get different frequencies. Notice that the range where you get a low firing frequency is very narrow.

7 Spike frequency adaptation

If continuously stimulated, a neuron tends to reduce its firing frequency after a few spikes. This effect is believed to be caused by the accumulation of calcium ions on the inside of the membrane. This calcium enters through specific voltage dependent calcium channels that open during the spikes much like the sodium and potassium channels. Calcium is a positive ion so the entering ions tend to depolarize the cell. However, calcium also influences other things in the cell.

Calcium has, among other things, the effect of opening calcium dependent potassium channels ($K_{(\mathrm{Ca})}$ channels), distinct from the ordinary voltage dependent potassium channels. When the $K_{(\mathrm{Ca})}$ channels open, they tend to hyperpolarize the cell, making it harder to activate.

In the model, we represent the intracellular calcium level with a single variable. Calcium channels are similar to the sodium and potassium channels but we also monitor the amount of calcium entering. The $K_{(\mathrm{Ca})}$ channels in the model simply have a conductance proportional to the calcium level. This is a rough approximation but fulfills its present purpose.

7.1 Calcium channels

- Start from the file `calcium` which describes one neuron "Control" which is the same cell as you used above (with sodium and potassium channels), and one neuron "N1" which also has voltage dependent calcium channels.
- Simulate. Did the calcium channels influence the spike frequency at all?
- Change the stimulation parameters to give a 5 ms pulse of 5 nA current. This should initiate one single spike which makes it possible to see what happens after the spike, the afterpolarization.
- Simulate. Do you see any effect from the calcium current? Is it depolarizing or hyperpolarizing?

7.2 Calcium dependent potassium channels

- Add another neuron "N2" with a calcium pool and calcium dependent potassium channels added. This is done by including the text "KCa" in the list of included properties.
- Do you see any difference in the afterpolarization?
- Change the stimulation back to continuous stimulation. Vary the current and notice that it is now possible to get low frequency firing in a much broader range of current values.
- Does the frequency stay the same during a continuous current stimulation? (Check for both high and low frequencies).
- Add a line " `fast Plot=Yes`" to plot the intracellular calcium level too (note that the line should be indented). Notice how the calcium level builds up at the beginning of the burst.

8 Synaptic interaction

Conventional chemical synapses are modeled as conductance increases in the postsynaptic (receiving) cells membrane. When a spike is produced in a presynaptic cell, and after a specified delay, a short pulse of increasing conductance is generated in the postsynaptic compartment. The shape of this pulse in these examples is modelled as a sudden increase in conductance followed by an exponential decay back to zero. Such conductance pulses can be seen as *postsynaptic potentials* in the soma.

Different types of synapses (excitatory/inhibitory) are permeable to different types of ions. This corresponds to different equilibrium potentials for the currents passing through. Excitatory synapses are permeable to primarily sodium and potassium giving a net equilibrium potential around 0 mV. Inhibitory synapses, on the other hand, are mainly permeable to chlorine ions, giving them an equilibrium potential around −85 mV. Excitatory postsynaptic potentials (*EPSP*s) are normally seen as positive pulses while inhibitory postsynaptic potentials (*IPSP*s) are normally negative.

One type of excitatory synaptic receptors has some special properties which makes them especially interesting. These receptors are known as NMDA-receptors and are known to be involved in synaptic plasticity. They are blocked by magnesium ions at normal resting potential levels. This has the consequence that presynaptic action potentials alone are not sufficient to get an effect on the postsynaptic side. To have an effect, the receiving cell already has to be somewhat depolarized. In the simulator, this voltage dependence is modeled with the same formalism as used for the voltage dependent channels.

8.1 Postsynaptic potentials

- Look at the file `synapse`. This specifies three neurons. One acts as the presynaptic neuron while the other two receives input through two excitatory synapses. Note that the synapses are situated on compartment `<1 1>` which is halfway out on the dendritic tree.
- Look at the EPSPs. Move the synapse to other compartments and try to see some difference.

8.2 Temporal summation

- Put the synapse on compartment `<1>`.
- Change the current to 5 nA. This should be enough to make the postsynaptic cell reach its spike threshold.

Notice that the EPSPs accumulate so that the potential finally reaches the threshold and the cell fires. This is what is often referred to as *temporal summation* of synaptic input.

8.3 Spatial summation

Now we will switch to a situation where we have several presynaptic neurons giving synaptic input on different parts of the dendritic tree of one postsynaptic neuron. With this setup, we should be able to use the combined effect of the different synapses to reach the threshold. This is what is called *spatial summation*.

- Look at the file `more-synapses`. The variable `individual` (user defined) is used to make the presynaptic cells fire asynchronously.

- Notice that it is now possible to reach the threshold even though the synapses are located further out on the dendritic tree.

8.4 Inhibition

- Look at the file `inhibition`. This specifies two presynaptic cells, one inhibitory and one excitatory, and one common postsynaptic cell. (The graph for membrane potential of the neuron "Post" has been magnified five times.)
- Look at the EPSPs and IPSPs. Note that the IPSPs are hardly visible. Why? (The synaptic conductances are equal and so is the distance from the soma).
- Raise the current to 5 nA, and move the synapse from neuron "Ex" to neuron "Post" to compartment `<1 1 1>`. Notice that the inhibitory synapse is on `<2 1>` (see figure 5.2).
- Look at the result in the postsynaptic cell.
- Now move the inhibitory synapse to `<1 1>`. Do you see any difference? Explain!

References

[1] Ö. Ekeberg, P. Hammarlund, B. Levin, and A. Lansner. SWIM — A simulation environment for realistic neural network modeling. In J. Skrzypek, editor, *Neural Network Simulation Environments*, pages 47–71. Kluwer, Hingham, MA, 1994.

[2] Ö. Ekeberg, M. Stensmo, and A. Lansner. SWIM — A simulator for real neural networks. Tech. Rep. TRITA-NA-P9014, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, 1990.

A The files you are supposed to modify

This appendix contains listings of the files used in the "hands on" examples in this tutorial. You are supposed to use these files as starting points for the experiments you are supposed to do. Note that most of the detailed parameters are taken from a separate file called "`default-cell.spec`" which is listed in appendix B.

The files can be found in `/info/biomod02/lab`**3**/.

A.1 The file "passive"

This file is used in section 5.

```
include "default-cell.spec"

neuron N (Passive)
  -- Give a 100 ms current pulse
  -- to the soma compartment
  <soma>
        Stim = yes
        Stim
          I = 0.0e-9    -- Injected current
          Start = 0.000 -- Start time
          Stop = 0.100  -- Stop time
end
```

A.2 The file "spikes"

This file is used in section 6.

```
include "default-cell.spec"

-- This is the stimulation parameters,
-- common for both neurons
define Stimulation
  <soma>
        Stim = yes
        Stim
          Start = 0.100
          Stop = 0.400
          I = 1e-9
end

-- This is the first neuron
neuron N1 (Passive_Tree Stimulation)
end

-- This is the second neuron
neuron N2 (Passive_Tree Na Stimulation)
end
```

A.3 The file "calcium"

This file is used in section 7.

```
include "default-cell.spec"

-- This is the stimulation parameters
define Stim
  <soma>
        Stim = yes
        Stim
          Start = 0.100
          Stop = 0.400
          I = 1.0e-9
end

-- This is the reference neuron,
-- do not change it
neuron Control (Passive_Dend Na K Stim)
end

-- This is the neuron to change
neuron N1 (Passive_Dend Na K Ca Stim)
end
```

A.4 The file "synapse"

This file is used in section 8.

```
include "default-cell.spec"

-- The presynaptic cell
neuron Pre (Passive_Tree Na K Ca KCa)
  <soma>
        Stim = yes
        Stim
          Start = 0.100
          Stop = 0.400
          I = 1e-9
end

-- First postsynaptic cell
neuron Post1 (Passive_Tree Na K Ca KCa)
end

-- First synapse
synapse Pre - Post1 (Excitatory_Synapse)
  compartment = <1 1>
end

-- Second postsynaptic cell
neuron Post2 (Passive_Tree Na K Ca KCa)
end

-- Second synapse
synapse Pre - Post2 (Excitatory_Synapse)
  compartment = <1 1>
end
```

A.5 The file "more-synapses"

This file is used in section 8.3.

```
include "default-cell.spec"

-- Parameters common for
-- all presynaptic cells
Define Pre (Passive_Tree Na K Ca KCa)
  <soma>
        Stim = yes
        Stim
          Start = 0.010 * individual
          Stop = 0.400
          I = 2e-9 * (1 - 0.1 * individual)
end

-- The postsynaptic cell
neuron Post (Passive_Tree Na K Ca KCa)
end

neuron Pre1 (Pre)
  individual = 1
end

synapse Pre1 - Post (Excitatory_Synapse)
  compartment = <1 1>
end

neuron Pre2 (Pre)
  individual = 2
end

synapse Pre2 - Post (Excitatory_Synapse)
  compartment = <1 2>
end

neuron Pre3 (Pre)
  individual = 3
end

synapse Pre3 - Post (Excitatory_Synapse)
  compartment = <2 1>
end

neuron Pre4 (Pre)
  individual = 4
end

synapse Pre4 - Post (Excitatory_Synapse)
  compartment = <2 2>
end
```

A.6 The file "inhibition"

This file is used in section 8.4.

```
include "default-cell.spec"

-- The presynaptic cell
Define Pre (Passive_Tree Na K Ca KCa)
  <soma>
        Stim = yes
        Stim
          Start = 0.010 * individual
          Stop = 0.400
          I = 1e-9 * (1 + 0.1 * individual)
end

-- The postsynaptic cell
neuron Post (Passive_Tree Na K Ca KCa)
end

neuron Ex (Pre)
  individual = 1
end

neuron Inh (Pre)
  individual = 2
end

synapse Ex - Post (Excitatory_Synapse)
  compartment = <1 1>
end

synapse Inh - Post (Inhibitory_Synapse)
  compartment = <2 1>
end
```

B Default specifications

This file contains default definitions used in all the examples in this tutorial. You do not have to read or understand this to do the examples. This file is only included here for reference.

```
---------- Passive properties ----------
define Passive
  E
    initial = -0.07
    plot
      min = -0.1
      max = 0.05
  <soma> E Plot = yes

  Eleak = -0.070
  Gm = Cm/Tconst
  Cm = area * 0.01
  Gcore = 6 * area

  <soma>
    area = (Diam*Diam*3.14)

  <...>
    area = <soma>area * areaD_S

  Tconst = 0.020
  <soma> Diam = 44e-6
  <...> areaD_S = 4.0
end

define Passive_Dend (Passive)
  <1> = Yes
  <1 1> = Yes
  <1 1 1> = Yes
end

define Passive_Tree (Passive)
  <1> = Yes
  <1 1> = Yes
  <1 1 1> = Yes
  <1 1 2> = Yes
  <1 2> = Yes
  <1 2 1> = Yes
  <1 2 2> = Yes
  <2> = Yes
  <2 1> = Yes
  <2 1 1> = Yes
  <2 1 2> = Yes
  <2 2> = Yes
  <2 2 1> = Yes
  <2 2 2> = Yes

  <*> areaD_S *= 0.5
  <* ...> areaD_S *= 0.25
end

---------- Active properties ----------

-- Sodium channels --
define Na
  <soma> Na = yes
  Na
    alpha
      A = 0.2e6
      B = -45e-3
      C = 1e-3
    beta
      A = .06e6
      B = -54e-3
      C = 20e-3
    exponent = 3
    initial = 0.0
    plot
      min = 0.0
      max = 1.0

    inact
      alpha
        A = 0.08e6
        B = -45e-3
        C = 1e-3
      beta
        A = .4e3
        B = -41e-3
        C = 2e-3
      exponent = 1
      initial = 1.0
      plot
        min = 0.0
        max = 1.0

    G = 167 * area
    E = .050
end

-- Potassium channel --
define K
  <soma> K = yes
   K
     alpha
       A = 0.020e6
       B = -45e-3
       C = 0.8e-3
     beta
       A = .005e6
       B = -35e-3
       C = 0.4e-3
     exponent = 4
     initial = 0.0
     plot
       min = 0.0
       max = 1.0

     G = 83 * area
     E = -0.080
end

-- Calcium channel --
define Ca
  <soma> Ca = yes
  Ca
    alpha
      A = 0.080e6
      B = 15e-3
      C = 11e-3
    beta
      A = 0.001e6
      B = 15e-3
      C = 0.5e-3
    exponent = 5
    initial = 0.0
    plot
      min = 0.0
      max = 1.0

    G = 5 * area
    E = 0.150
  fast rate = 0.67e12 * area
end

-- Calcium dependent potassium channels --
define KCa
  <soma> KCa fast = yes

  -- Fast Ca pool --
  fast
    decay = 20
    initial = 0.0
    plot
      min = 0.0
      max = 10.0

  KCa
    fast
      E = -0.080
      G = 40e-9

  -- Slow Ca pool --
  slow
    decay = 2
    initial = 0.0
    plot
      min = 0.0
      max = 3.0
```

```
  KCa
    slow
      E = -0.080
      G = 40e-9
end

-- NMDA channels --
define NMDA
  NMDA
    alpha
      A = 0.7e3
      C = 17e-3
    beta
      A = 1.8*5.6
      C = 17e-3
    exponent = 1

    initial = 0.0
    plot
      min = 0.0
      max = 1.0

    ext
      E = 0.000
      G = 75 * area * bathnmda
      slow E = 0.020
      slow rate = 1.107e9 * NMDA ext G

  bathnmda = 0
end


------------ Synapses------------

-- The basic synapse --
define Default_Synapse
  open plot
      min = 0.0
      max = 1.0

  delay = 0.001
  duration = 0.001
  raisetime = 0.0005
  decaytime = 0.020
  pre = <soma>
end

-- The standard excitatory synapse --
define Excitatory_Synapse (Default_Synapse)
  E = 0.000
  G = 40e-9
  compartment = <1 1>
end

-- The standard inhibitory synapse --
define Inhibitory_Synapse (Default_Synapse)
  E = -0.085
  G = 40e-9
  compartment = <1>
end

-- The standard NMDA containing synapse --
define NMDA_Synapse (Excitatory_Synapse)
  NMDA = yes
  compartment = <soma>
  G = 0
  decaytime = 0.100
  NMDA G = 4e-9
  NMDA E = 0.000
  NMDA slow rate = 1.107e9 * NMDA G
  NMDA slow E = 0.020
end
```