
NEURON - tutorial D of Gillies & Sterratt (part 1)

<http://www.anc.ed.ac.uk/school/neuron/>

COSC422 – lecture 10

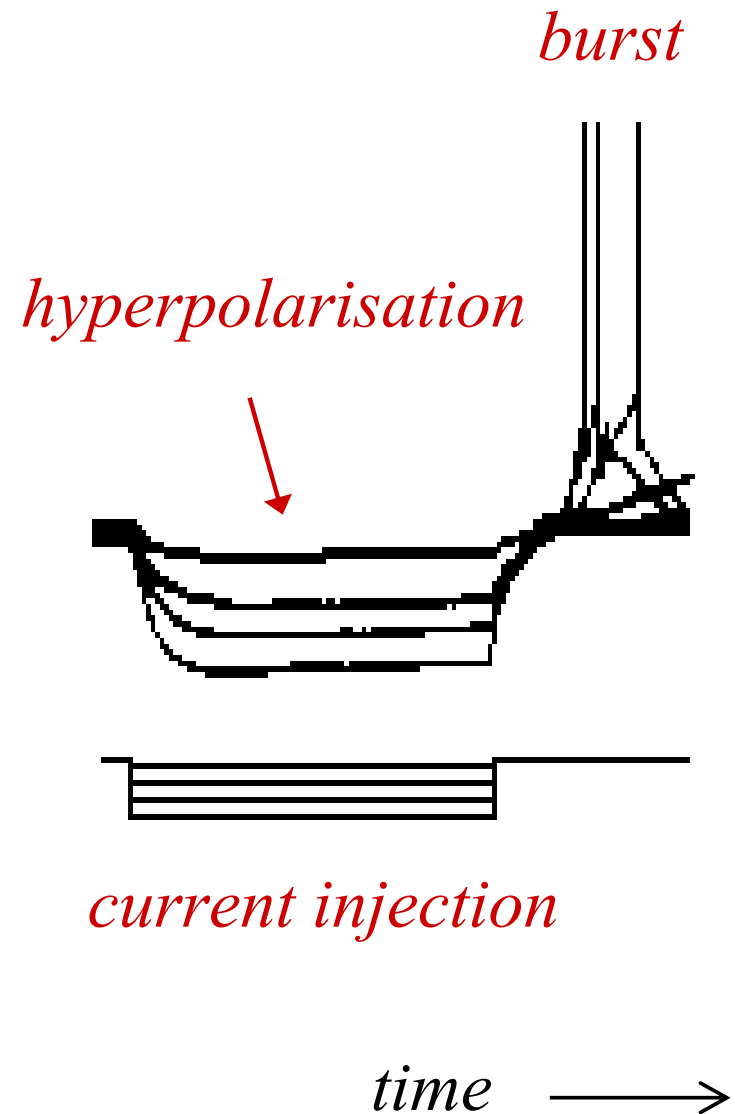
How to program ion channels with **NMODL**

Introduction: our current goal

- So far our simulated neurons were given a characteristic morphology for a subthalamic nucleus projection neuron.
- But they still contain only the default Hodgkin and Huxley types of sodium and potassium ion-selective channels.
- We would like to make these neurons much more electrophysiologically similar to subthalamic nucleus neurons.
- There are many channel types we would want to add, but for now we will add only one new type, called the T-type calcium channel.

Post hyperpolarising response

- A typical electrophysiological feature of subthalamic neurons is the post hyperpolarising response.
- When a neuron is hyperpolarised (e.g. by current injection or inhibitory synaptic input), at the end of the hyperpolarisation, a burst of spikes is observed in subthalamic projection neurons (see right).
- This response is mediated by a low threshold calcium selective ion channel, called the **T-type Ca channel**.



T-type calcium electric current

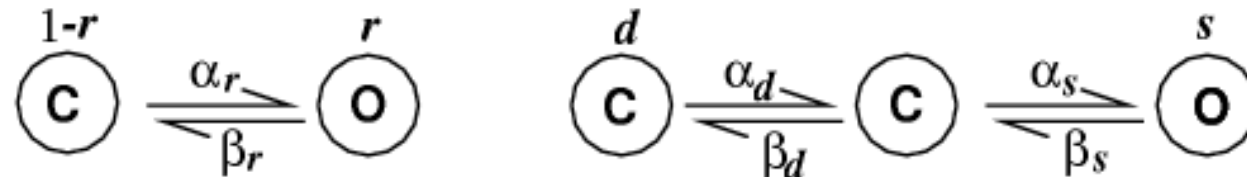
- First we must know the equation that describes the electric current that flows through the T-type Ca-channel.
- The current I_T produced from the T-type Ca-channels was characterised within the Hodgkin-Huxley framework by Wang et al. (J. Neurophys. 66: 839-850, 1992):

$$I_T = g_{T(max)} r^3 s (V - E_{Ca})$$

- where $g_{T(max)}$ is the maximum T-type Ca conductance; r is the **activation** state variable; s is the **inactivation** state variable, E_{Ca} is the reversal potential for Ca; V is the neuron membrane potential

T-type calcium channels state variables

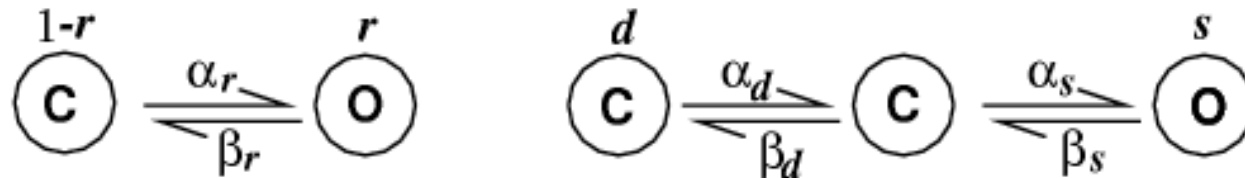
- State variables r and s are given by the following kinetic systems:



- Here "C" refers to a closed state and "O" refers to an open state.
- The α and β denote the forward and backward rate constants from one state to another; they are voltage dependent functions specified by Wang et al. (1992).
- Note that inactivation is a three state kinetic process, with fast (s) and slow (d) components.

T-type calcium channels state variables

- State variables r and s are given by the following kinetic systems:



- These kinetic schemes translate to three differential equations:

$$\dot{r} = \alpha_r(1-r) - \beta_r r$$

$$\dot{s} = \alpha_s(1-s-d) - \beta_s s$$

$$\dot{d} = \alpha_d(1-s-d) - \beta_d d$$

T-type calcium channels: rate functions α & β

- Rate functions obey these equations (J. Neurophys. 66: 839-850):

$$\alpha_r = 1 / \left(1.7 + e^{-\frac{V+28.2}{13.5}} \right)$$

$$\beta_r = e^{-\frac{V+63.0}{7.8}} / \left(1.7 + e^{-\frac{V+28.2}{13.1}} \right)$$

$$\alpha_s = e^{-\frac{V+160.3}{17.8}}$$

$$\beta_s = e^{-\frac{V+160.3}{17.8}} \left(\sqrt{0.25 + e^{-\frac{V+83.5}{6.3}}} - 0.5 \right)$$

$$\alpha_d = \left(1.0 + e^{-\frac{V+37.4}{30.0}} \right) / (240.0 (0.5 + d)) \quad \beta_d = \alpha_d (d - 0.5)$$

$$d = \sqrt{0.25 + e^{-\frac{V+83.5}{6.3}}}$$

NEURON model description language

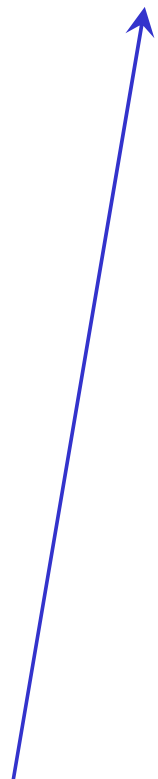
- We would like to add this channel to our model. This cannot be done with the programming language **hoc** that we have used in the previous tutorials.
- Instead we have to learn the **NEURON Model Description Language (NMODL)** provided for defining additional membrane mechanisms such as ion channels or ion pumps or point processes such as synapses.
- There is more written by Mike Hines and Ted Carnevale at:
<http://www.neuron.yale.edu/neuron/papers/nc2000/nmodl.htm>

CaT.mod

- A membrane mechanism description using NMODL is laid out in an ordinary text file. The text file **CaT.mod** containing a specification of the T-type Ca channel in NMODL looks like this:

```
TITLE Calcium T channel for
Subthalamic Nucleus
UNITS {
    (mV) = (millivolt)
    (mA) = (milliamp)
}
NEURON {
    SUFFIX CaT
    USEION ca READ eca WRITE ica
    RANGE gmax
}
PARAMETER {
    gmax = 0.002 (mho/cm2)
}
```

```
ASSIGNED {
    v (mV)
    eca (mV)
    ica (mA/cm2)
    ralpha (/ms)
    rbeta (/ms)
    salpha (/ms)
    sbeta (/ms)
    dalpha (/ms)
    dbeta (/ms)
}
STATE {
    r s d
}
```



CaT.mod – contd.

```
BREAKPOINT {
    SOLVE states METHOD cnextp
    ica = gmax*r*r*r*s*(v-eca)
}

INITIAL {
    settables(v)
    r = ralpha/(ralpha+rbeta)
    s = (salpha*(dbeta+dalpha) - (salpha*dbeta)) /
        ((salpha+sbeta) * (dalpha+dbeta) - (salpha*dbeta))
    d = (dbeta*(salpha+sbeta) - (salpha*dbeta)) /
        ((salpha+sbeta) * (dalpha+dbeta) - (salpha*dbeta))
}

DERIVATIVE states {
    settables(v)
    r' = ((ralpha*(1-r)) - (rbeta*r))
    d' = ((dbeta*(1-s-d)) - (dalpha*d))
    s' = ((salpha*(1-s-d)) - (sbeta*s))
}
```

CaT.mod – finish

```
UNITSOFF
```

```
PROCEDURE settables(v (mV)) {  
    LOCAL bd  
    TABLE ralpha, rbeta, salpha, sbeta, dalpha, dbeta  
        FROM -100 TO 100 WITH 200  
  
    ralpha = 1.0/(1.7+exp(-(v+28.2)/13.5))  
    rbeta  = exp(-(v+63.0)/7.8)/(exp(-(v+28.8)/13.1)+1.7)  
  
    salpha = exp(-(v+160.3)/17.8)  
    sbeta  = (sqrt(0.25+exp((v+83.5)/6.3))-0.5)*(exp(-(v+160.3)/17.8))  
  
    bd      = sqrt(0.25+exp((v+83.5)/6.3))  
    dalpha  = (1.0+exp((v+37.4)/30.0))/(240.0*(0.5+bd))  
    dbeta   = (bd-0.5)*dalpha  
}
```

```
UNITSON
```

NEURON and NMODL

- A membrane mechanism description using NMODL is laid out in an ordinary text file.
- Suppose we have created a text file **CaT.mod** containing our description of the T-type Ca channel in NMODL and we want to insert it into the soma.
- We simply do it like this (see **sthD.hoc**):

```
soma {  
    nseg = 1  
    diam = 18.8  
    L = 18.8  
    Ra = 123.0  
    insert hh  
    gnabar_hh = 0.25  
    gl_hh = .0001667  
    el_hh = -60.0  
    insert CaT  
}
```

Note on equilibrium potentials

- The value of **eca** is calculated by NEURON. How and when it is calculated depends on the particular mechanisms dealing with ions that are inserted in a section (i.e., soma or dendrites).
- NEURON uses a built-in, i.e. "default", **eca**. But NEURON's default values for **eca**, **ena**, and **ek** are not appropriate for our mammalian subthalamic nucleus.
- We want to use typical mammalian values, which are
 - **ena** = 71.5 mV;
 - **ek** = -89.1 mV;
 - **eca** = 126.1 mV

New SThcell template

- We will set all of these equilibrium potentials as our model has channels using each of these ion species.
- These should be set **after** the channel mechanisms are inserted into a section, and *must be set for each section* that has channels using that ionic species inserted.
- We can modify our SThcell template, adding these equilibrium potential parameter values to the soma block:

```
soma {  
    nseg = 1  
    diam = 18.8  
    L = 18.8  
    Ra = 123.0  
    insert hh  
    ena = 71.5  
    ek = -89.1  
    gnabar_hh = 0.25  
    gl_hh = .0001667  
    el_hh = -60.0  
    insert CaT  
    eca = 126.1  
}
```

NEURON and NMODL: compilation

- Assume a membrane mechanism description using NMODL is laid out in the text file (e.g., the text file **CaT.mod** containing our description of the T-type calcium channel in NMODL).
- The NEURON interpreter cannot read this file directly as it can with **hoc** files. Instead, the NMODL file has to be **compiled** into a form that NEURON can use.
- How to compile & incorporate this new mechanism into NEURON depends on what operating system you are using:

Linux / Unix

- Change to the dir that contains the **CaT.mod** file, and there type the command
- **> nrnivmodl**
- This creates a new NEURON executable called **i386/special** whose library of mechanisms includes our T current (the i386 subdirectory will be created automatically if it does not already exist) .

Linux / Unix contd & MacOS

- To launch this new executable and have it load to your model file, at the OS prompt in the directory that contains **i386** just type
- **> nrngui sthD.hoc**
- **MacOS users** only need to drag the directory that contains **CaT.mod** and drop it onto the **mknrndll** icon. This produces a **nrnmech.dll** file that is automatically loaded when you double click on any **.hoc** file in the same directory.

MS Windows

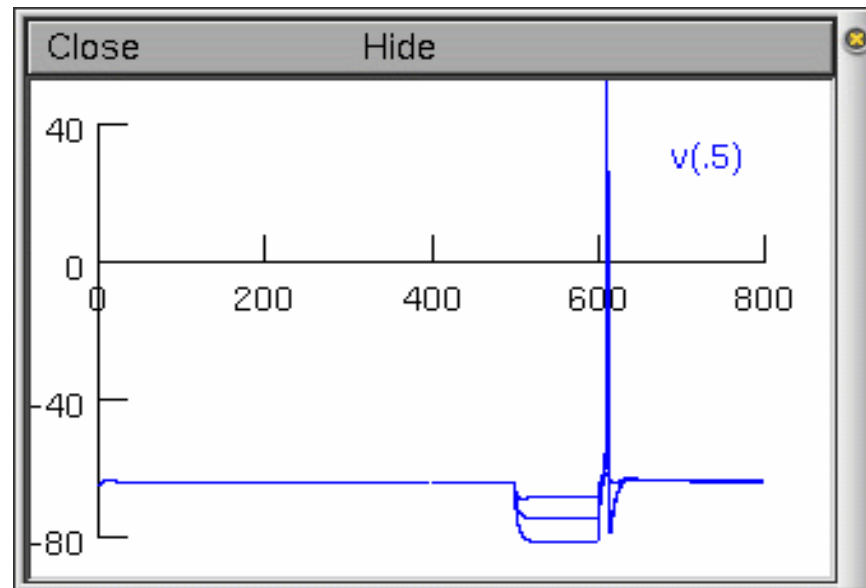
- You just launch
- > **Start/Programs/Neuron/mknrndll**
- This brings up a directory browser that can be used to navigate to your working directory that contains the **CaT.mod** file.
- When you get to the proper directory, click on the button labelled "**Make nrnmech.dll**". This compiles all the **.mod** files in this directory and creates a file called **nrnmech.dll** that contains the new compiled mechanisms.
- **nrnmech.dll** will be automatically loaded when you double click on a **.hoc** file in this directory.

Why the compilation?

- Why do we have to go through this cumbersome compilation procedure every time we want to create or modify a channel?
- The reason is that membrane mechanisms are used on every time step, and therefore need to be efficient.
- Converting the NMODL file to C-code and then compiling this into a new NEURON program or library (which is what **nrnivmodl** and **mknrndll** do) leads to more efficient simulation.

Conclusion

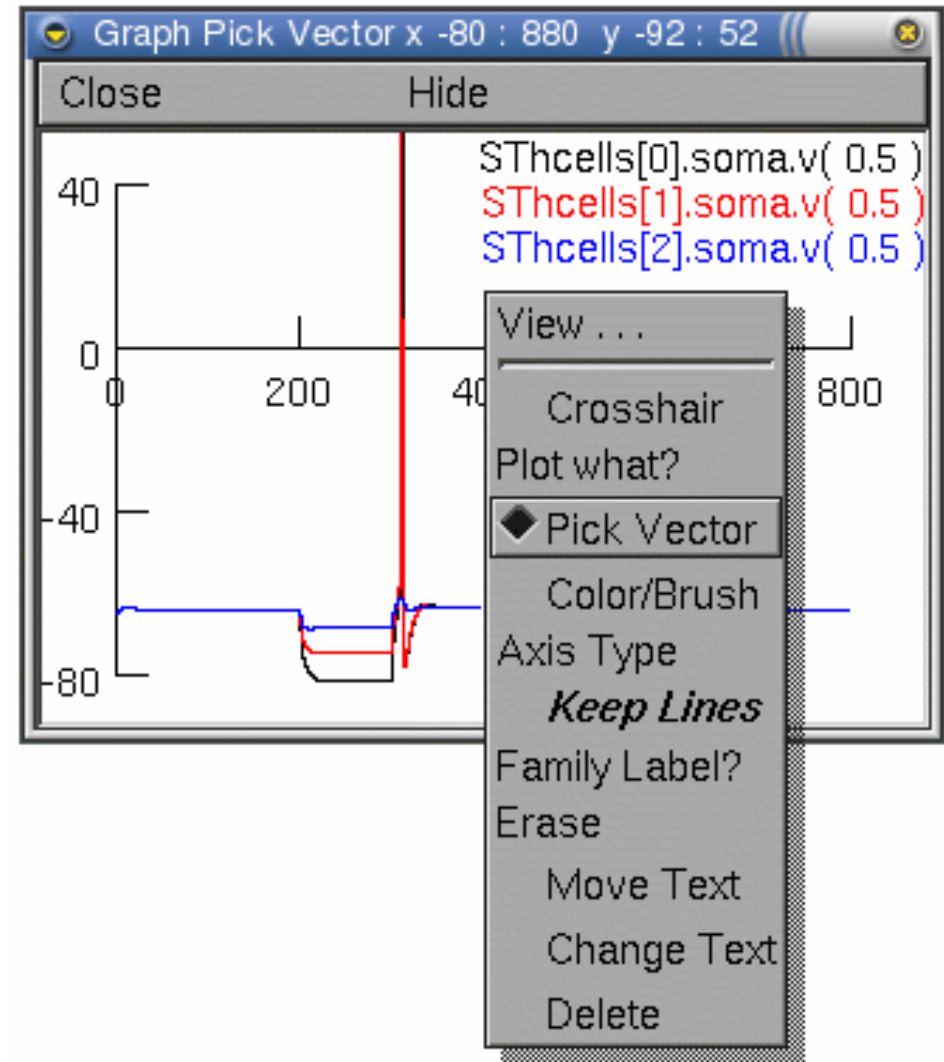
- Now if we launch **sthD.hoc** with a hyperpolarising current injected into any of our neurons, we now observe a post-hyperpolarising T-type response:



- With current injections of -0.1, -0.2, and -0.3 nA.
- Note: The action potential is unrealistic for mammalian subthalamic cells b/c it is still based on HH squid axon channels.

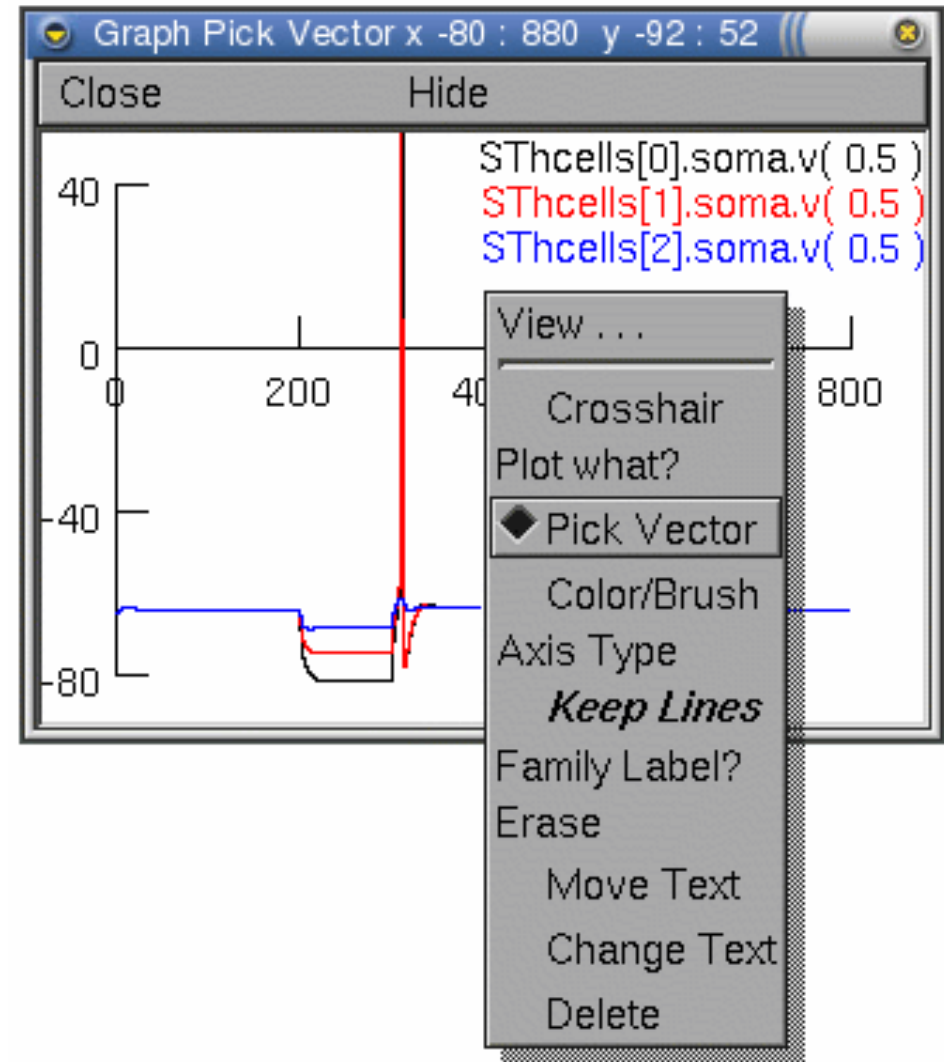
Picking vectors from the plots

- Everything that is plotted can be saved to a specified data filename as the numerical values of a plotted variable.
- For example, if we are plotting the voltage of the subthalamic neuron and want to record a particular voltage trace as numbers, we can do this from the *Graph Properties* menu. This menu appears after a right click on the graph of interest.



Picking vectors from the plots

- From this menu, select the *Pick Vector* option.
- Any plot on this graph can be selected simply by clicking on the particular plotted line of interest.
- In our example, the graph contains three plots (showing the voltage at the soma of three subthalamic cells). **We can only select one plot at a time to save!**



Saving the vectors in a file **name.dat**

- Click on the voltage trace you want to save as numbers and go to the NEURON *Main Menu* toolbar.
- The *Main Menu* toolbar contains a *Vector* menu allowing vectors to be saved or retrieved from data files.
- Select *Save File* from the *Vector* menu.
- This pops up a file dialogue window allowing you to enter a filename for saving the selected vector data.

Structure of the vector data file

- The beginning of the data file looks something like this:

```
label:SThcells[2].soma.v( 0.5 )
32001
0      -65
0.025 -65.0488
0.05   -65.0844
0.075  -65.1125
0.1    -65.136
```

- The first line of the file is a text string identifying the data being plotted, the second line is the number of data points in the file (here 32001), and finally the third line onward contains the data. First column is the time and, second column, soma voltage at that time.