

1 Contents

Auckland Bioengineering Institute CellML tutorial	0
1 Contents	1
Auckland Bioengineering Institute CellML OpenCell Tutorial.....	2
2 About this Tutorial	2
3 Modifying Hodgkin & Huxley 1952 toward Noble 1962	2
3.1 Hodgkin-Huxley Model Structure.....	3
3.2 Hodgkin-Huxley Model Validation.....	3
3.2.1 Potassium Channel Kinetics Validation	4
3.2.2 Action Potential Validation	5
3.3 Modification toward Noble 1962 Model	6
3.3.1 Examining the Mathematical Model.....	6
3.3.2 Modifying the Parameters.....	7
4 Ten Tusscher, Noble, Panfilov 2004 (TNNP) stimuli.....	8
4.1 Periodic.....	8
4.2 S1-S2 Restitution.....	9
5 Conclusion.....	10

Auckland Bioengineering Institute CellML

OpenCell Tutorial

All material is copyright Auckland Bioengineering Institute, unless otherwise stated. Please see www.cellml.org for copyright and licensing statements.

2 About this Tutorial

This is a tutorial for CellML emphasizing the modularity, reuse, and exchange features that are available when developing models in a CellML 1.1 framework. The focus of this tutorial is the application of a CellML based modelling framework in the area of cardiac cellular electrophysiology.

The reference description can be browsed from the models.cellml.org a1 workspace:
<http://models.cellml.org/workspace/a1/>

There are two parts to the tutorial. The first is a detailed guide to the modification of an existing “published” model which is available on the internet. The example used is modifying the original Hodgkin & Huxley 1952 squid axon model to produce action potentials closer to those found in the Noble 1962 model of cardiac cells.

The second part applies the same techniques to a more realistic scenario, namely that of modifying the stimulus protocol being applied to a human cardiac myocyte model.

For this tutorial, you will be provided with a folder containing the files required to complete the exercises.

Basic usage of the OpenCell software is provided in a built-in manual; this will explain how to perform basic tasks such as creating new elements in the tree view, or managing models and clones. Many of the editing functions in OpenCell are carried out through the use of the context menu in the tree view or graphing pane.

3 Modifying Hodgkin & Huxley 1952 toward Noble 1962

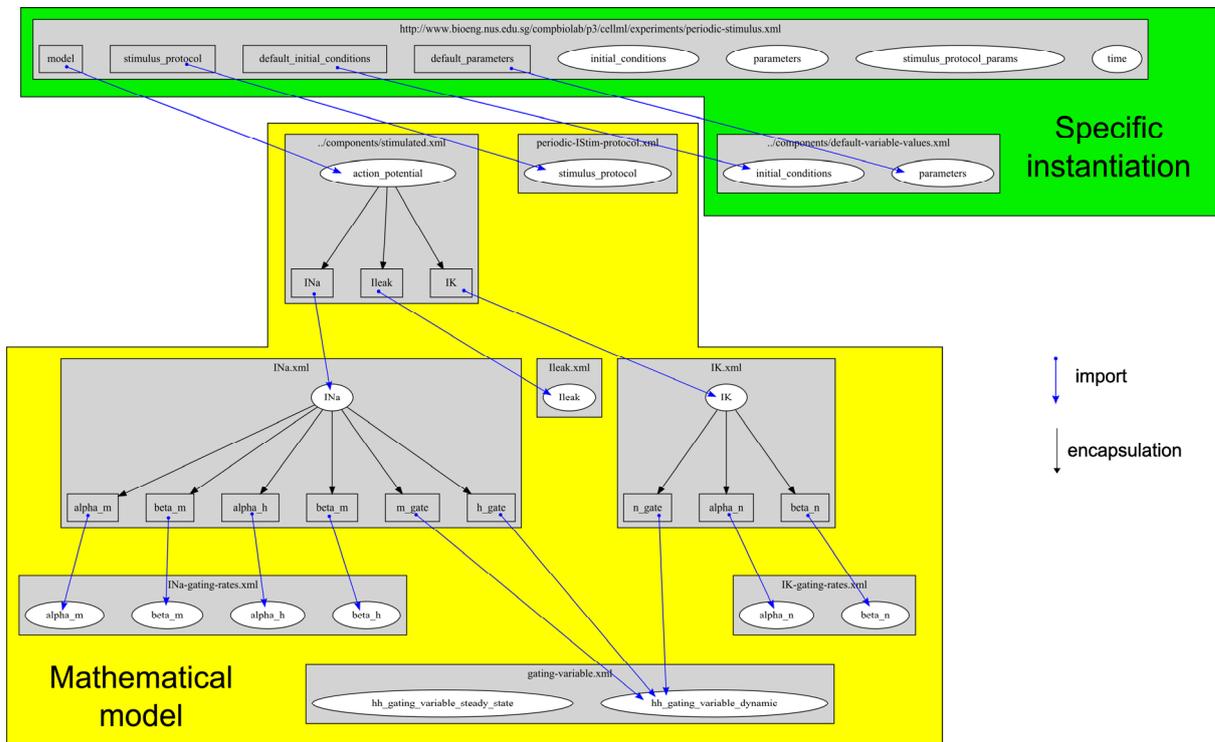
This section of the tutorial illustrates the modularity and reuse aspects of CellML 1.1. An accurate representation of the original Hodgkin & Huxley 1952 squid axon model has previously been published and made available on a web server at NUS in Singapore.

In this task we first validate this implementation of the model and then modify it through the replacement of the potassium current from a model available in the CellML model repository, and the slight modification of some parameters. These modifications are sufficient to turn the short duration of a nerve action potential into a long lasting cardiac action potential similar to the Noble 1962 cardiac Purkinje fibre action potential.

The basic premise of this example is based on the first biophysical cardiac cellular electrophysiology model, published by Denis Noble in 1962. The Hodgkin-Huxley 1952 squid axon model is modified, such that it becomes capable of reproducing the action and pacemaker potentials observed in Purkinje fibres. In this tutorial we start with a CellML description of the Hodgkin-Huxley model and modify it to reproduce some of the key features of the Noble 1962 model.

3.1 Hodgkin-Huxley Model Structure

The structure of the NUS implementation of the Hodgkin-Huxley model is illustrated in the figure below:



This diagram emphasizes the modular definition of the model using CellML 1.1. The mathematical model is separated from the instantiation into a specific simulation experiment. Reuse is illustrated by the use of a single definition of the standard Hodgkin-Huxley gating kinetics being reused for each of the m, h, and n gating variables.

3.2 Hodgkin-Huxley Model Validation

The Hodgkin-Huxley model used here has been previously implemented and is available online from the Computational Bioengineering Laboratory at the National University of Singapore. The original source of these models is <http://www.bioeng.nus.edu.sg/combiolab/p3/>. All the graphs illustrated here are simulated directly from the data described on the NUS server. The following simulations serve to validate this implementation by showing simulation observations which correspond directly to those published by Hodgkin & Huxley (1952).

All references to files loaded in OpenCell are relative to the IUPS-2009-CellML-tutorial-models folder extracted from the model archive (ROOT).

We describe below the steps for recreating two of the original validation studies provided under <http://www.bioeng.nus.edu.sg/combiolab/p3/>.

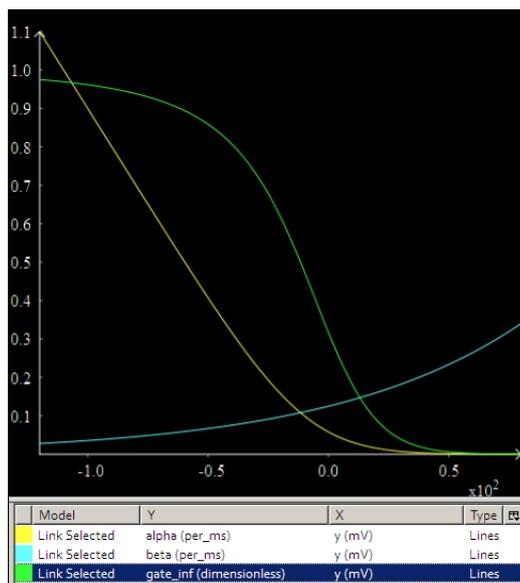
3.2.1 Potassium Channel Kinetics Validation

We will perform a validation study, investigating the kinetics of the potassium channel of the base implementation of the Hodgkin-Huxley model. Start OpenCell and load the model:

```
ROOT/NUS-HodgkinHuxley-Implementation/experiments/IK-gating-kinetics.xml.
```

The simulation metadata gets parsed (this sets up the settings in the *integration settings* panel) so you can leave all of the integration settings as they are.

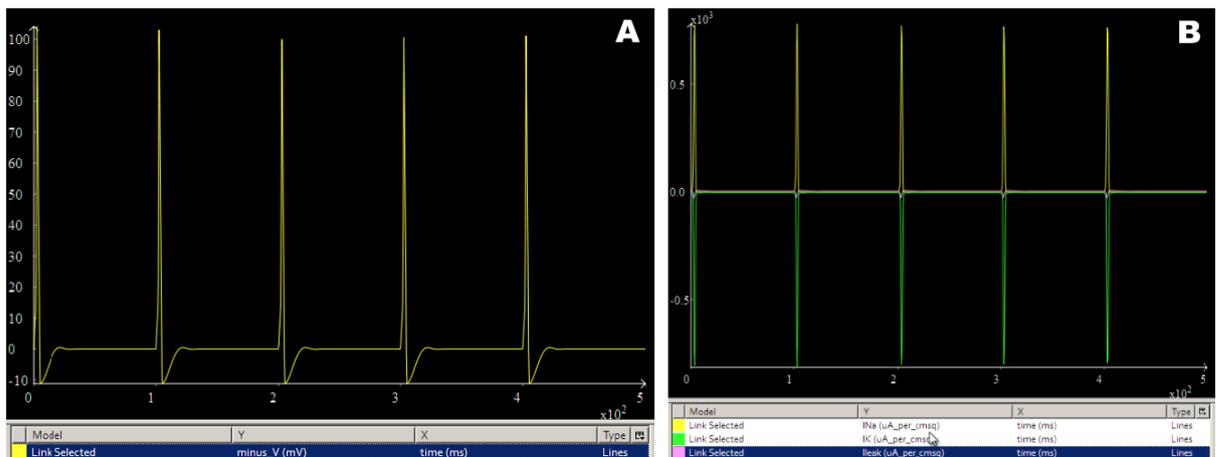
1. Click the Integrate button, and wait until the Integration Complete message appears.
2. Plot α_n and β_n against *membrane potential*. To do this, right click in the area below the graph panel, and select *New Trace* from the context menu. Click where it says *click to select*, and select α_n/α for the Y axis, and *ramp/y* for the X axis (membrane potential is described in the model mathematically as a linear ramp with a generic name, although imported as V). This reproduces figure 4 from the Hodgkin & Huxley (1952) article, shown as the yellow and green traces in the graph below.
3. Plot n_{∞} against *membrane potential*. [*membrane potential* as above; $n_{\infty} = hh_gating_variable_steady_state/gate_inf$ (generic gating kinetics imported for each gate)]. This reproduces Figure 5 from the HH (1952) article, shown as the cyan trace below.
4. Click the large *Integrate* button near the bottom centre of the OpenCell window in the integration pane. The graph traces will appear as shown in the graph below, although the colours will be different as they are randomly selected. You can choose the colour of a trace by clicking on the small coloured box at the left of the trace settings, and selecting a new colour from the pop-up palette.



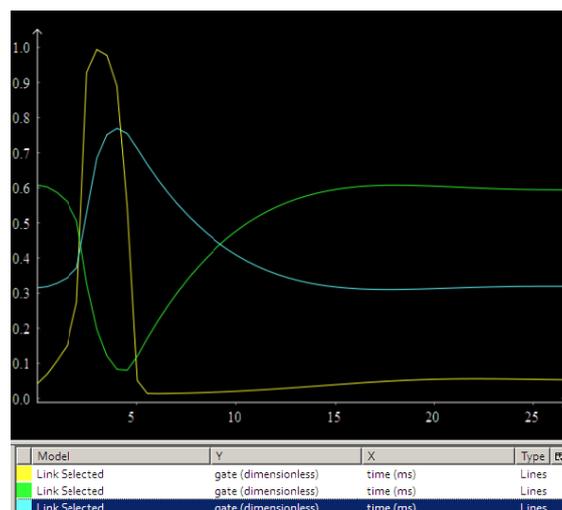
3.2.2 Action Potential Validation

We will perform a validation study investigating the dynamic response of the base implementation of the Hodgkin & Huxley model to an applied electrical stimulus current.

5. Load the following model into OpenCell:
 ROOT/NUS-HodgkinHuxley-Implementation/experiments/periodic-stimulus.xml into OpenCell.
 Once again, the simulation metadata gets parsed and you can just click Integrate.
6. Plot action potentials [$action_potential/minus_V$ vs $time/time$], as shown on graph A below.
7. Plot currents [INa/INa vs $time$; IK/IK vs $time$; $Ileak/Ileak$ vs $time$], as shown on graph B below. The first variable can be plotted simply by changing the Y variable to Ina/Ina for that trace.

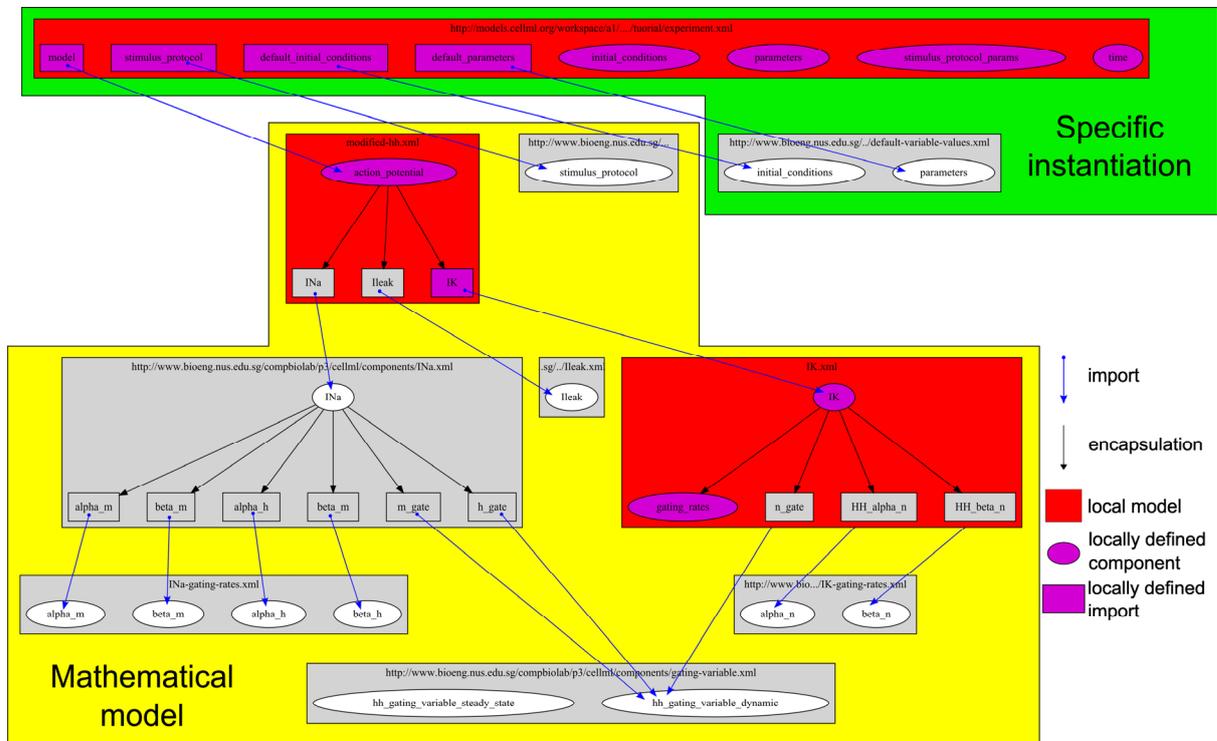


8. Plot gating variables [all three of $hh_gating_variable_dynamic/gate$ vs $time$]. The graph below shows the gating variables n (cyan), m (yellow), and h (green). You can do this either by changing the Y variable on all 3 traces, or deleting the traces and creating them from scratch. Use the right mouse button to zoom in on the graph in order to view a single action potential as shown below.



3.3 Modification toward Noble 1962 Model

This task defines the modifications made to the base HH implementation in order to approximate the action potential shape of the Noble 1962 model. The figure below illustrates the model structure that we want to develop:



In this figure, all the red models are defined “locally” and the rest come from the base implementation from NUS, as described above. Within the local models, some of the components are imported from the base NUS model and some are newly defined. All units are imported from the base NUS model implementation. The major change to the mathematical model is to replace the Hodgkin & Huxley potassium channel with one which is based on the potassium channel described by the Noble 1962 model. To achieve this, the Noble potassium channel description has to be altered to reflect the modelling conventions used by Hodgkin & Huxley.

3.3.1 Examining the Mathematical Model

Load the following model into OpenCell:

```
ROOT/models.cellml.org/workspace/a1/tutorial/modified-hh.xml
```

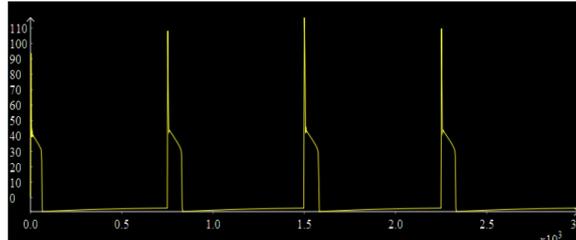
Ignore the message that says the model is unable to be integrated, and select the "show complete model structure" tree view by clicking on the second icon on the tree view toolbar. You can now see the structure of the actual mathematical model. Take a look at the imports elements, showing that the sodium and leakage current are being imported from “NUS” and the potassium current is being imported locally.

3.3.2 Modifying the Parameters

Close and re-start OpenCell. Now load the following model into OpenCell:

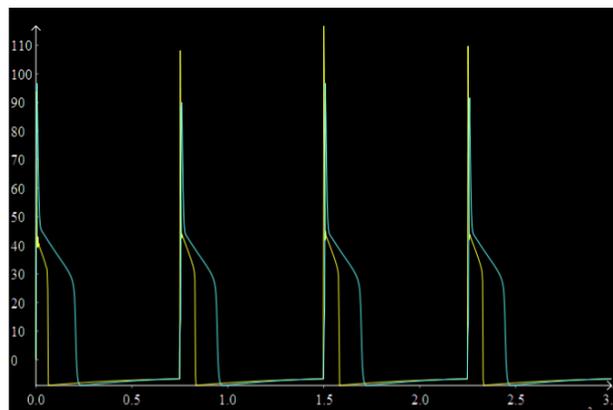
```
ROOT/models.cellml.org/workspace/a1/tutorial/experiment-start.xml
```

In this example, the modified mathematical model is imported into a specific experiment model and all the default parameter values and initial conditions are imported from the NUS model implementation. The stimulus protocol parameters are defined locally to define a suitable stimulus protocol. Integrating this model and plotting [*action_potential/minus_V vs time*] shows the altered action potentials which result from the changes made to the potassium current formulation. At this point, click on the trace settings where it says *Link selected*, and change this to *Copy selected model*. This will make this trace keep the original model data once we create a clone.



We will now make the changes required to create the plateau-like phase in the action potential.

9. Clone the model by using the context menu on the model in the listing at the far left, and selecting Clone. Select the new cloned model, which will be named with a timestamp.
10. Create a new component in the tree view (bring up the context menu on the Components element icon and select New). Name this component `parameters`.
11. Create a new variable in the parameters component, called `Cm`. Give it a value of `12`, and units of `uF_per_cmsq` (predefined in the drop-down menu). Change the public interface to `out`.
12. Create another new variable in the parameters component, called `gK_max`. Give it a value of `12`, and units of `mS_per_cmsq`. Change the public interface to `out`.
13. Expand the `model, default_parameters` connection, and delete the variable mappings for `Cm` and `gK_max` - these will be the first two, but you must expand them to see what the variable mapping actually is.
14. Create a new connection (use the context menu on the `Connections` element) and connect the components `model` and `parameters` using the drop-down selection boxes.
15. Create two new variable mappings within this new connection using the context menu on the `model, parameters` element. Expand the two variable mappings. Select `Cm` in both `variable 1` and `variable 2` for the first mapping, and select `gK_max` for both in the second mapping.
16. Integrate your edited model, and graph the same variables [*action_potential/minus_V vs time*] as for the original model. You should notice that the action potentials have the plateau-like phase, as shown below on the cyan trace:



4 Ten Tusscher, Noble, Panfilov 2004 (TNNP) stimuli

Here we make use of an existing encoding of the ten Tusscher et al (2004) human ventricular myocyte model which has a periodic stimulus current applied. We replace this stimulus protocol with one suited for restitution analysis.

Nickerson & Buist (2008) (PBMB, Practical application of CellML 1.1...) published a study investigating the practical application of CellML 1.1 based on the ten Tusscher et al (2004) human ventricular myocyte electrophysiology model. The online supplement for this study is available at:

<http://www.bioeng.nus.edu.sg/compbiolab/papers/nickerson-PBMB-2008/>

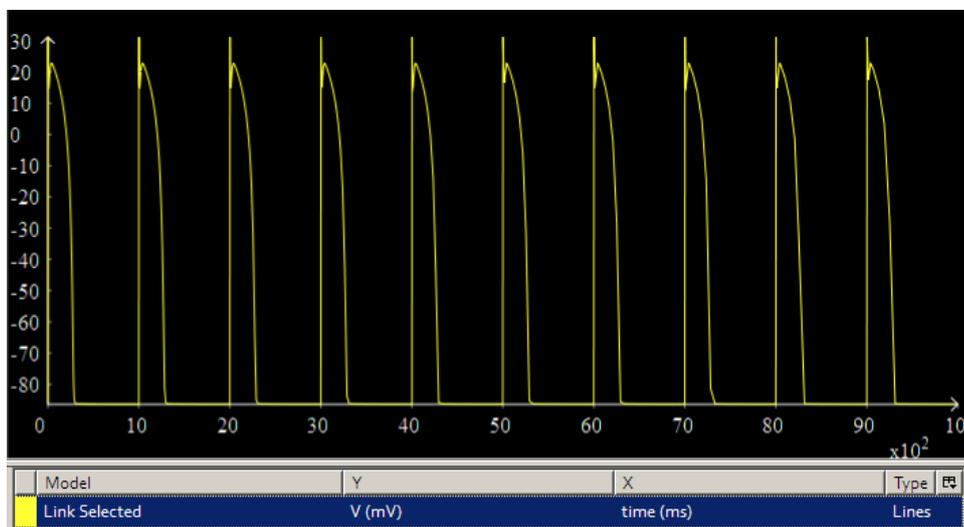
This includes all the simulation experiment descriptions presented in that study. As for the Hodgkin-Huxley example above, we first check that the base implementation available from the NUS server is correct and then modify it with a new stimulus protocol.

4.1 Periodic

Close and re-start OpenCell. Now load the following model into OpenCell:

```
ROOT/NUS-TNNP-Implementation/papers/nickerson-PBMB-2008/experiments/TNNP-periodic.xml
```

17. Change the end time point to **10000 ms** in the integration pane. Turn on tabulation control with a tabulation interval of **1ms** - the tabulation settings are in the *Advanced* settings tab of the integration pane.
18. Create a trace for the action potentials; *membrane_potential/V* vs *time/time*. The list of components and their variables in this model is very long, and you will have to scroll the list in order to find the ones you need to graph.
19. Click the *Integrate* button.



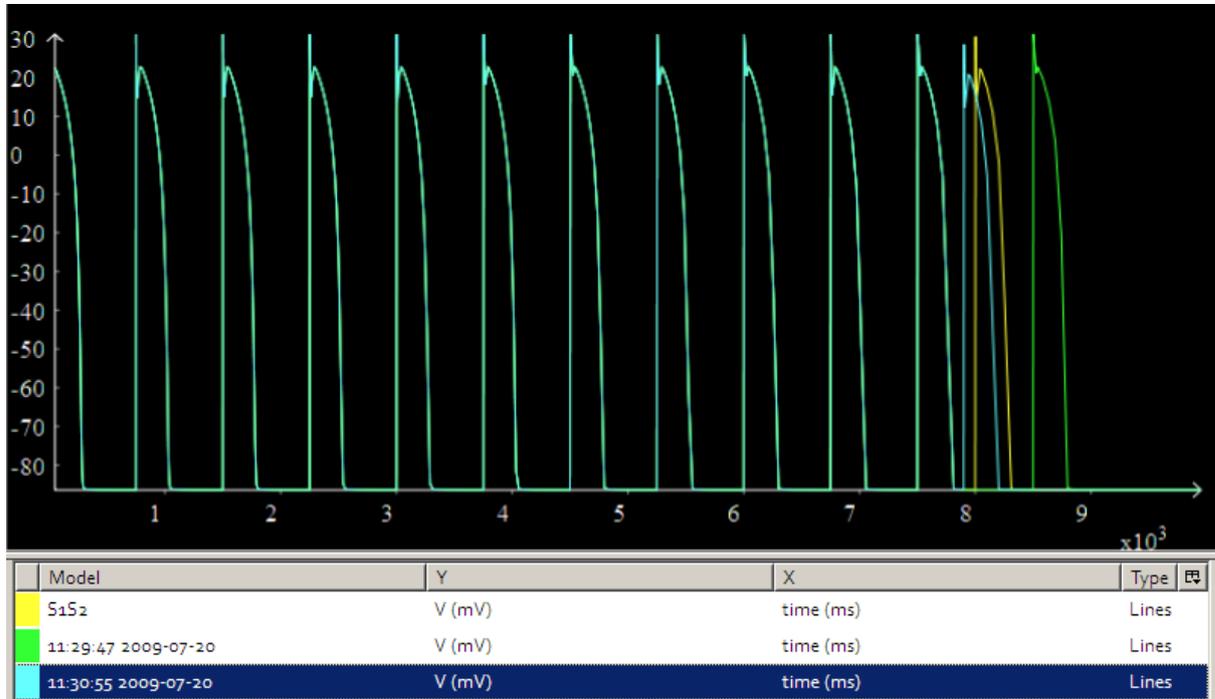
4.2 S1-S2 Restitution

The Nickerson & Buist (2008) implementation of the ten Tusscher et al (2004) model includes the description of the model with an applied S1-S2 stimulus protocol (10 stimuli with a fixed period followed by a single stimulus at some interval after the 10th periodic stimulus. This is a published and “fixed” set of CellML encoded models which we are not able to change directly, but we would really like to use this model with our own stimulus protocol description. Due to the modular nature of the model implementation available from NUS this is relatively straightforward to achieve, by following these steps:

1. Create a new model in OpenCell by selecting *New* from the *File* menu, or by clicking on the *New model...* button on the toolbar. Select CellML 1.1 as the version, and give the model the name **S1S2**.
2. Save the model to **ROOT/my-S1S2.xml**.
3. Create a new import, and enter the following in the text field:
NUS-TNNP-Implementation/papers/nickerson-PBMB-2008/experiments/TNNP-S1S2.xml
4. Create a new component in this import; enter *model* and *interface* into the two text fields.
5. Create another new import, and enter the following in the text field:
URL: **NUS-TNNP-Implementation/cellml/models/common/units.xml**
6. Create a new *units* within this import; enter *ms* and *ms* into the two text fields.
7. Save your model again.
8. Create a new component in the top level model, with the name **parameters**.
9. Create a new variable within this new component, using the following settings:
name: **basicCycleLength**
value: **750**
units: *ms*
public_interface: *out*
10. Create another new variable within the parameters component, using the following settings:
name: **S2Interval**
value: **500**
units: *ms*
public_interface: *out*
11. Save your model again.
12. Create a new connection, expand it, and select parameters and model for the two components to connect.
13. Create new variable mapping within this new connection, and select *basicCycleLength* for *variable 1* and *variable 2*.
14. Create another new variable mapping, and select *S2Interval* for *variable 1* and *variable 2*.
- 15. Save your model again!**
16. Right click on your model in the model listing at the far left and select *Reload*. This should now enable the integration of the model.
17. Set the tabulation control to on using the check box (in the *Advanced settings* tab), and set a tabulation interval of **1ms**.
18. Set the integration end time to **10000ms**, and the maximum step size to **0.1ms**.
19. Integrate the model, and plot the action potentials by creating a trace for *membrane_potential/V vs time/time*.
20. Change the trace setting from *Link selected* to *Copy selected model*.
21. Right click on the model in the model listing and select *Clone*. Select this cloned model.
22. Edit the *S2Interval* variable value of this cloned model; change it to **1000** for example - now integrate the cloned model, and create a trace of *membrane_potential/V vs time/time*.

23. Repeat steps 20, 21, and 22, but this time set the *S2Interval* to 400.

You should now have a graph that shows three action potential traces - the first 11 peaks should overlap, and the last should be different between the three traces. Below is a graph showing these three traces, where *S2Interval* has been set to 400ms (cyan), 750ms (yellow), and 1000ms (green).



5 Conclusion

In this tutorial we have introduced you to the basic concepts of CellML 1.1, in particular the modularity and re-use features of the language. For further tutorials and information, please visit the *getting started* section at <http://www.cellml.org/getting-started/>.

We encourage you to get involved with the CellML project via the mailing lists and the tracker. You will find links to the mailing lists on the *getting involved* section of the website. You can discuss features of and issues with the software on the CellML sections of the Physiome Tracker, at <https://tracker.physiomeproject.org/>.