MASTERS IMMERSION IN
NEURAL NETWORK SIMPLIFICATION

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Semester Project Report (2016)

Neuron Model Simplification: a NEST model perspective

Michel Fabrice Jaquier

December 26, 2016

Report Type:	Final	
Masters Student:	Michael Jaquier	....
Project Director:	Dr. Marc-Oliver Gewaltig	....
Date:	December 26, 2016	

1 Work Objectives

Compartmental neuron models are computationally expensive , necessitating a large investment for even moderately sized simulations. Model simplifications is one approach by which computationally expensive simulations can be faithfully reproduced on hardware orders of magnitude less powerful. We seek to facilitate this speedup by porting models written for NEURON into NEST where they may be simulated as generalized integrate and fire (GIF) neurons.

We face three major hurdles towards accomplishing the stated goal. First, compartmental models respond to stimulation in significantly differing fashion. Establishing a universal stimulation current which provides the necessary range of responses is intractable. Second, extracting the necessary parameters from the high fidelity models provided in NEURON format depends on the usage of high-throughput fitting toolboxes. Third, the sheer quantity of neurons – 31,346 neurons spread out among 207 morphoelectrical types – demands the use of automated work-flow practices.

We address these hurdles through a mixture of automation techniques, exploiting a minimum of *a priori* knowledge, maximizing specific subthreshold parameters from the NEURON models, and integrating fitting methods from the toolbox GIFFittingToolbox [2].

The goals of this work are as following:

- Understand and implement an appropriate spike generating current into a arbitrary NEURON model to investigate the full range of neuron dynamics.
- Fit the NEURON spiking data to a simplified GIF model.
- Extract the minimum necessary parameter space from the GIF model to build a functional NEST model.
- Streamline and automate the simplification process from NEURON –> GIF –> NEST. For any arbitrary NEURON model.

2 Work Achieved in the Past Semester

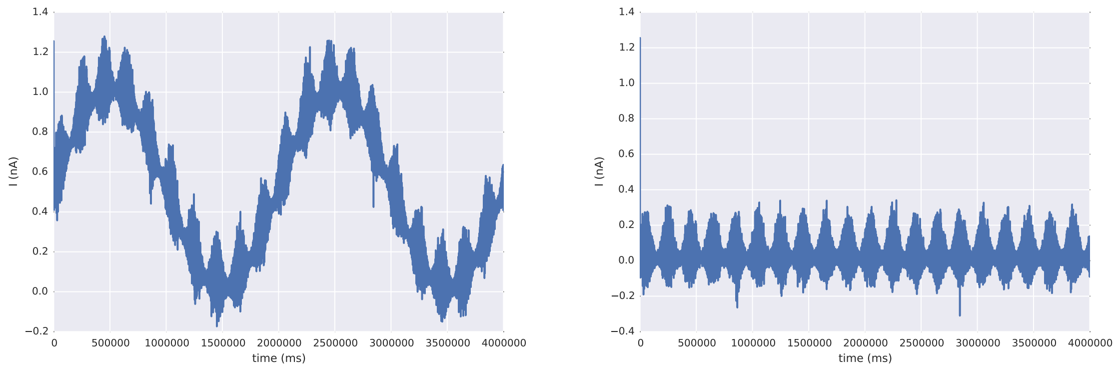
2.1 Stimulation Protocol

Models simulated in NEURON are capable of accepting current vectors containing all simulation data; this feature can be exploited to generate neuron firing patterns amenable to the fitting procedure utilized. We once more stress the intractability of the problem if approached with the hope of utilizing the same current vector across all potential NEURON models.

A modified Ornstein-Uhlenbeck process is utilized to generate a current which is highly malleable:

$$I_{t+1} = I_t \times \sigma_0[1 + \Delta\sigma \times \sin(2\pi f t)] + \mu \tag{1}$$

```
def generate_current(time_):
current = moving_current[0]
yield current
current = ou_process[index] * variance[index] + moving_current[index]
```



(a) (b)

Figure 1: (a) Current generated from modified Ornstein-Uhlenbeck process with time dependent mean. (b) Current generated from modified Ornstein-Uhlenbeck process without time dependent mean

The above simplified code block allows us to generate a current, which when tuned may be applied to any reasonable NEURON model.

The noise portion of the stimulus applied to the NEURON model is tuned by observing the sub-threshold variance as suggest by *Pozzorini et al* [2]. Sub-threshold variance is adjusted to a lower boundary of $3mV$ and an upper boundary of $8mV$. Bounds are then applied to the following equation to define the variance:

$$\sigma_0 = \frac{\sigma_0^{max} + \sigma_0^{min}}{2}. \tag{2}$$

$$\Delta\sigma = \frac{\sigma_0^{max} - \sigma_0^{min}}{2\sigma_0}. \tag{3}$$

To ensure a full sweep of the parameter space – investigating all possible modes of adaption – the current is further modulated by the inclusion of a time dependent mean (fig. 1(a)). Modulating the current in such a way expands the range of neurons the protocol functions with.

Variance, and the time dependent mean (μ) can be modified to accept a range of different functions. In 1(a) both parameters are modified by a sin wave with a frequency determined by the experimental conditions. For a 100s training set and 20s test set a frequency of 0.2Hz would be optimal [3].

We modify the mean to drive the model through a range of spiking behavior between 0-13Hz. A brute force method is applied to solve for optimal μ stimulating the NEURON model over one full period of the sin wave; assessing the maximum and minimum firing frequency for optimal μ.

The resultant current (fig 2) explores the entire parameter space.

2.2 Fitting the NEURON model

Fitting was accomplished with the help of the GIFFITTINGTOOLBOX , a toolbox written in python designed to fit arbitrary neuron models and generating GIF models parameter sets. The fitting toolbox provides an invaluable tool for generating GIF models so long as the input is carefully sanitized.

GIFFITTINGTOOLBOX offers a host of different methods and filter to fit the data to a GIF model, for the following we utilize a rectangular filter with the following meta parameters:

```
length=500.0,
binsize_lb=2.0,
binsize_ub=100.0,
slope=4.5
```

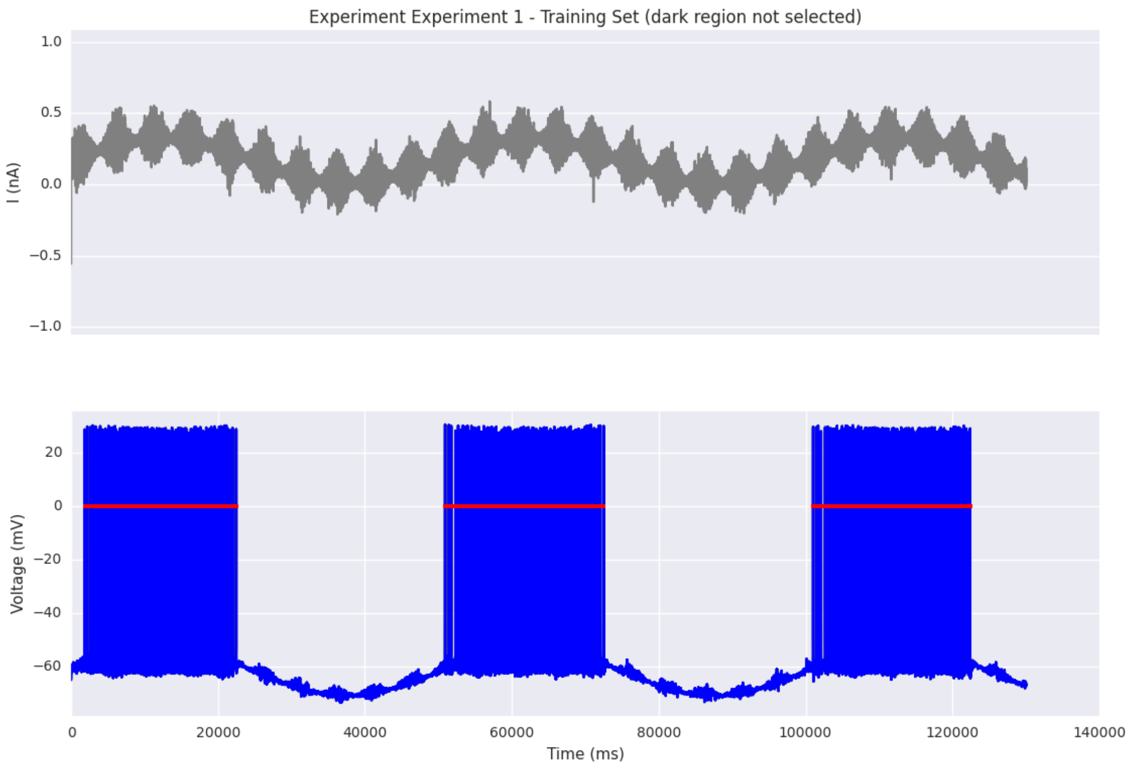


Figure 2: **Top** – Current input. **Bottom** – Spiking behavior of the L5 TTPC1 cADpyr232 1 NEURON model generated from the modified UO process.

The GIFFITTINGTOOLBOX has the following limits to the input space. Failure to meet the limiting criteria will negate any guarantees of convergence.

- Sub-threshold dynamics must show some variance.
- The spiking behavior must not be in the saturated region during the entire training period.
- Sufficient spikes must be detected in the training period (3-13Hz).
- The training period must be of sufficient length (100s-120s is sufficient).
- The test period must be of sufficient length (10s-20s is sufficient).

Details on GIFFITTINGTOOLBOX [2] can be found in the wiki (http://wiki.epfl.ch/giftoolbox). We will not cover the implementation of fitting further, interested readers are invited to explore the wiki for details. We will focus attention instead to the components specifically useful for extraction of NEST GIF models.

The toolbox provides several metrics to assess goodness of fit. Of which we will utilize three to determine the fidelity of fitting.

- ΔV – Relative stochasticity of model fit
- $\frac{Bit}{Spike}$ – Maximized Log-Likelihood [1]
- %variance explained – The model variance explained relative to test set(s).

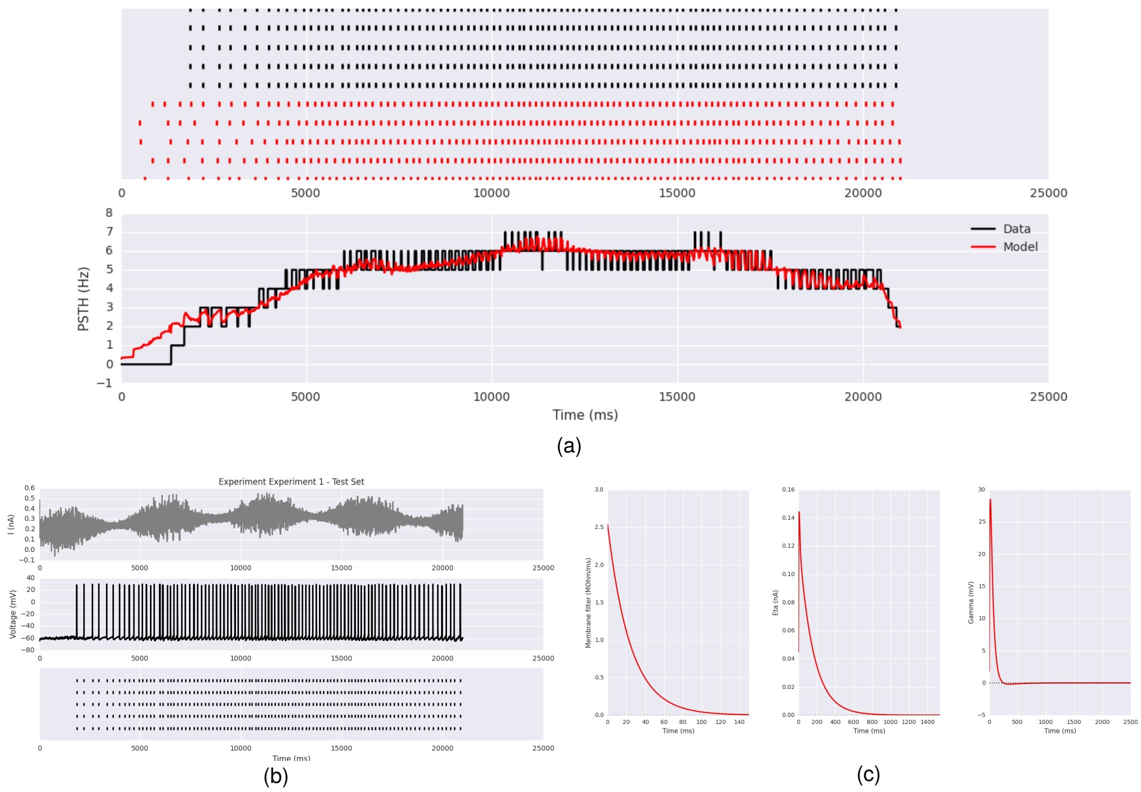


Figure 3: (a) Model data compared to actual Testing data. **top** Raster plot of test data (red) and training data (black). **bottom** Spiking frequency of input testing data (black), and model spiking frequency (red). (b) 20s of prototypical model testing input **top** Current vector. **center** Voltage trace. **bottom** Raster plot of spiking activity (c) **left** Membrane filter. **center** Eta (η) the spike-triggered current. **right** Gamma (γ) the spike-frequency adaption parameter

The fitting procedure is straight forward. A 120s long current is injected into the NEURON model (fig 2. The resultant membrane potential together with the injected current are loaded into the GIFFITTINGTOOLBOX . Time constants related to the spike triggered current (η) and spike frequency adaption (γ) are automatically fit, prior to fitting the full model.

The model (fig 3(b)) is tested against five separate test sets generated by repeated 20s current injections (fig 3(a) *top*). After generating the model and testing against the test set the model is accepted or rejected based on goodness of fit as determined by the three above parameters. Values will depend upon the model chosen.

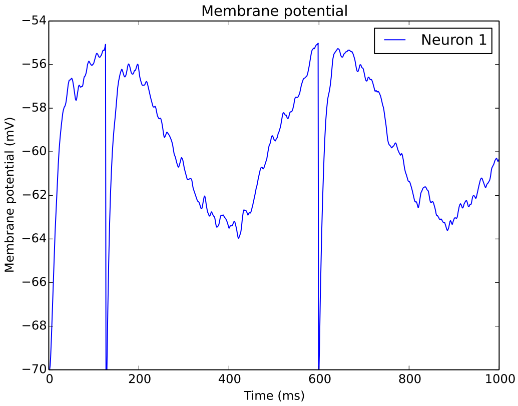
Care should be taken in selecting a range of ΔV. Stochastic models will naturally have a much higher value relative to deterministic models. Further we utilize % variance explained as a metric over the available M_D^* [4] which proves a more reliable metric for stochastic neuron models.

2.3 Parameter Extraction for NEST

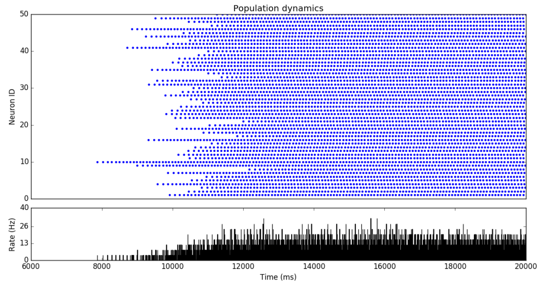
Parameters are extracted from the GIF with minor modification:

$$\gamma_{\text{NEST}} = \frac{\gamma}{1 - \exp(\frac{T_{ref}}{\gamma})} \tag{4}$$

We apply the same modification to η in order to correct for GIFFITTINGTOOLBOX ignoring data near spikes. A minor correction is applied to the membrane capacitance to convert from GIFFITTINGTOOLBOX providing values in nS and the nest model requesting pS.



(a)



(b)

Figure 4: (a) Membrane voltage trace for GIF model simulated in NEST. (b) 20s of spiking activity for a cluster of 50 NEST GIF neurons.

The example dictionary of values utilized in the NEST simulation of a GIF neuron below, is the basis for the construction of the model presented in in fig 4. Where *q-stc*, and *q-sfa* are the spike induced current and adaption which represent the neurons effective adaption potential. The time courses for each are extracted by repeated fittings of the GIFFITTINGTOOLBOX generated model.

```
'C_m': 393.72300433915924,
'Delta_V': 0.50293054308940099,
'E_L': -71.44614414609228,
'V_T_star': -59.111196470716493,
'V_reset': -60.180934363278432,
'g_L': 0.016232839466640048,
'lambda_0': 1.0,
'q_sfa': -156.06779567322599, 706.74848657739881, -64.73987636004604,
'q_stc': -36.113852013445381, 36.111325253486832, 4.9907522904942239,
't_ref': 4.0,
'tau_sfa': 10.0, 50.0, 250.0,
'tau_stc': 5.3229597950581473, 5.3573467026471668, 153.39658582308675
```

3 Current State of the Work

Currently the main objectives of the project have been carried out. We have demonstrated the viability of an automated fitting process to refactor computationally expensive NEURON models to the GIF model simulated on the NEST platform.

4 Prospects for Future Work

- GIF models implemented in NEST are simple single-exponential models. It may provide valuable to expand those models to multi-exponential models.
- Additional automatic model verification between NEST and NEURON models would benefit the fitting process.
- Increased automation of data collection from NEURON.

5 References

[1] L. Paninski, J. Pillow, and E. Simoncelli, “Comparing integrate-and-fire models estimated using intracellular and extracellular data,” *Neurocomput.*, vol. 65-66, pp. 379–385, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2004.10.032

[2] C. Pozzorini, S. Mensi, O. Hagens, R. Naud, C. Koch, and W. Gerstner, “Automated high-throughput characterization of single neurons by means of simplified spiking models,” *PLOS Computational Biology*, vol. 11, no. 6, pp. 1–29, 06 2015. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pcbi.1004275

[3] C. Rössert, C. Pozzorini, G. Chindemi, A. P. Davison, C. Eroe, J. King, T. H. Newton, M. Nolte, S. Ramaswamy, M. W. Reimann, M.-O. Gewaltig, W. Gerstner, H. Markram, I. Segev, and E. Muller, “Automated point-neuron simplification of data-driven microcircuit models,” *ArXiv e-prints*, Mar. 2016.

[4] S. Mensi, R. Naud, C. Pozzorini, M. Avermann, C. C. H. Petersen, and W. Gerstner, “Parameter extraction and classification of three cortical neuron types reveals two distinct adaptation mechanisms.” *J Neurophysiol*, vol. 107, no. 6, pp. 1756–1775, Mar 2012.