Calibration documentation

Dominik Schmidt

January 22, 2014

1 Introduction

The aim of this document is to describe each step of the calibration that is done in the cake module. Please note that this document is unfinished may change at any time. Also note that some of the features described here might not be implemented yet. ALSO note that until now, only LIF parameters are supported.

2 Parameter types

Since the hardware has different types of parameters, we need to define different parameter spaces and transformations. The aim of a calibration is to accurately determine these transformations.

We concluded that we should define four different parameter spaces (note that the two examples given for each parameter space are corresponding):

- Biological parameters (Bio)

  → pyNN parameters, for example v_rest or tau_m

- Hardware parameters (HW)

  → Parameters that are measured on the hardware, for example E_l or g_l

- Hardware control parameters (HC)

  → Parameters that control the hardware, for example E_l or I_gl

  → Found in pyhalbe.HICANN.neuron_parameters and .. .shared_parameters

- Floating gate values (DAC)

  → Discreet floating gate values between 0 and 1023.

  → Voltages are calculated by $DAC * \frac{1800mV}{1023}$

  → Currents are calculated by $DAC * \frac{2500nA}{1023}$

Calibration should be applied as the **last** step (to DAC values) in order to make it independend on the type of user input. Until now, calibtic only supports this last step and it does not properly support transformations between each of these parameter spaces. This missing feature will be implented in calibtic soon.

3 Calibration

Calibration steps are listed in the order in which they should be run. For each parameter, you will find a description of what it does, how it is measured, what the calibration does to it and everything else that is worth noting.

The run_calibration tool in the pycake/bin/ folder will calibrate all working parameters. To configure the calibration, you have to edit parameters.py to match whatever you want to calibrate or measure. There you can edit the calibrated ranges, step length and number of repetitions (which will increase accuracy but also linearly increase experiment time). After calibration of a parameter, you should keep that variable within the range in which you calibrated - outside of the range, the accuracy can decrease.

3.1 E_syni and E_synx

E_syni and E_synx are the synaptic reversal potentials. These voltages are applied at the OTA_0 in the synaptic input circuit (see figure 1). Their pyNN-counterparts are e_rev_I and e_rev_I.

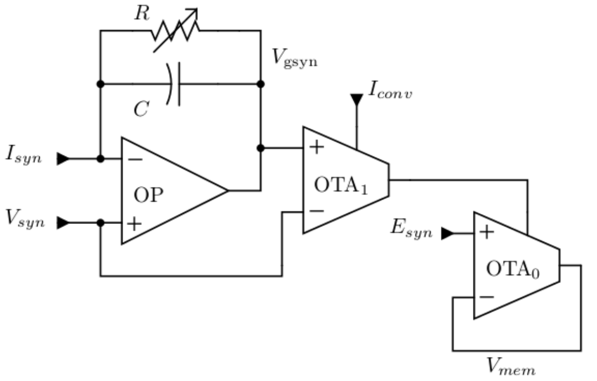

Figure 1: Synapse circuit. E_{syn} is applied to OTA1

Measurement

The synaptic reversal potentials are measured using leakage currents of the synaptic input circuit. Since inhibitory and excitatory input circuits are identical, I will only describe the measurement once.

Ideally, when no spike input is present, there should be no leakage current through OTA_0. However, due to imperfections in the manufacturing process, these leakages cannot avoided. Here, we make use of them by turning off all other conductances in the neuron circuit. With the synaptic input being the only remaining conductance, the membrane potential will approach the synaptic reversal potential. This can easily be measured with the ADC, thus giving an approximation of the reversal potential.

3.2 E_l

E_l is the membrane leakage potential that determines where the membrane potential should stay when no input is given at all. It is applied at the leakage OTA (see figure 2). The corresponding pyNN parameter is the resting potential V_rest.

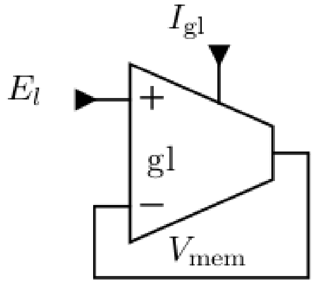

Figure 2: Synapse circuit. E_{syn} is applied to OTA1

While calibrating, the synaptic reversal potentials are set symmetrically around E_l. In all following calibrations, the synaptic reversal potentials are also set symmetrically around the calibrated E_l.

Measurement

E_l is measured by turning on all conductances that will later be turned on while experiments are run. That way, it is ensured that we calibrate an effective resting potential that balances the unwanted leakage currents. Then, without any input, the membrane potential approaches a value which will be considered as E_l.

3.3 V_t

V_t is the voltage at which a spike is initiated. This voltage corresponts to the pyNN parameter V_thresh in a LIF model.

Measurement

To measure V_t, the resting potential E_l is set above the threshold so that the neuron is spiking all the time. Then, the maximum of the trace is measured and considered to be V_t. There was a discussion whether it would be more exact to take the mean value of all the local maxima instead of one maximum for the whole curve. This idea was rejected mainly because this would underestimate V_t as the ADC does not always hit the exact maximum of the voltage trace, but measures mostly a little too early or a little too late.

3.4 V_reset

After a spike, the membrane potential is set to V_reset. This parameter corresponds to the pyNN parameter v_reset.

Since this parameter is a shared parameter, it can not be set for each neuron individually but only for blocks of 128 neurons. Thus, the calibration can not eliminate neuron-to-neuron variation but only shift the whole block in order to reach the correct mean V_reset.

To compensate for the deviation ΔV_{reset} of a neuron's individual V_reset from the desired V_reset of the whole block, all other voltages (E_l, E_syni, E_synx, V_t) of a neuron need to be shifted by ΔV_{reset}. Unfortunately, this ΔV_{reset} also depends (linearly) on the desired V_reset. The calibration first runs the calibration of the whole block as one experiment. After that, another experiment is run which determines the first order polynomials $\Delta V_{reset}(V\_reset)$ for each neuron. These polynomials are also saved in calibtic as the 21st parameter, although they are not a neuron parameter as defined by pyhalbe.HICANN.neuron_parameter module.

Measurement

V_reset is measured in the same way as V_t, except that the minimum of the trace is taken, not the maximum.