Migration of the Advanced Lab Course on Neuromorphic Computing

Alexander Nock
Supervisor: Christian Mauch

October 2020

1 Abstract

The aim of the internship was to translate the first two parts of the advanced lab course on neuromorphic computing for physics students onto the BrainScaleS-2 setup. This included the characterization of the membrane voltage which yielded nearly the same linear translation of V_{leak} and V_{reset} for DAC values over 550. In contrast, the dynamic range of V_{th} has been shown to be in the lower half of the DAC values, but sharing the same range of values for the membrane voltage. Furthermore, it could be shown that the intra-block deviation is significant and makes the trial-to-trial variation negligible.
The subsequently conducted calibration of the leak conductance yielded an improvement that is limited by the trial-to-trial variation and required varying the proportionality parameter which sets the strength of the correction and thereby accounting for a non-oscillating adaption of the leak conductance.

2 Introduction

The advanced lab course on neuromorphic computing is designed to provide an introduction for physics students to how neural networks operate and what they are capable of. Emulating biological neural networks yields a higher power efficiency compared to ordinary computers and also an increased processing speed in specific tasks like image recognition. By building neuromorphic hardware, it is possible to gain the same results as with just simulating the behavior of spiking neural networks without the effort of solving sophisticated differential equations.

The first task (of seven) in the lab course deals with the characterization of a single neuron ("Task 1: Investigating a Single Neuron") while the second task concentrates on the calibration of the leak conductance g_{leak} which is closely linked to the membrane time constant τ_m ("Task 2: Calibrating Neuron Param-

eters"). The scripts for those tasks (which are adapted to the new generation of hardware) can be accessed in the according directory[1].

2.1 Hardware of BrainScaleS-2

The basic model used to electronically describe the basic dynamics of neurons is the leaky integrate-and-fire model (LIF). Mathematically this is expressed by the following differential equation for the membrane voltage V_m:

$$C_m \frac{dV_m}{dt} = g_{leak}(V_{leak} - V_m) + \sum_i g_i^{syn}(t)(V_{syn} - V_m) \tag{1}$$

There the voltage V_m over the membrane capacitance C_m is pulled towards the leak voltage V_{leak} over time. This happens by current flowing through the leak conductance g_{leak}. Additionally, synaptic input which can either be excitatory or inhibitory modulates the membrane voltage and depends on the conductance $g_i^{syn}(t)$ of the synapse connection. For the internship the last term with the sum won't be considered since no synaptic input is applied. Actually, the BrainScaleS-2 design uses a more advanced model which is based on the LIF model and is called adaptive exponential integrate-and-fire (AdEx) model. It has additional terms to emulate the biological behavior of neurons more precisely. However, the LIF model will be sufficient for the course of this report.
The hardware that will be used for the advanced lab course in the future is based on the BrainScaleS-2 chip that has 512 neurons [3] in total. Those neurons are distributed over four equally sized blocks which have own memories storing the analog parameters for their 128 neurons (cf. figure 1) and therefore potentially lead to different values for neurons on different blocks even if all neurons are called with the same set of parameters. The chip has 256 synapses per neuron which yields a total of 512 x 256 = 131 072 possible connections. In contrast, the current chip used for the lab course is called Spikey and only has 384 neurons in total. However, the main difference between the two chip generations is the Plasticity Processing Unit (PPU) that is exclusively on the BSS-2 setup. It is a CPU with a vector processor which for example can be used to calculate the weight updates for several neuron connections simultaneously. Important to mention is that the neurons themselves show fixed pattern noise due to their manufacturing process which mainly results from differences in the size and dotation of the transistors.
The communication between the host computer and the chip works via a Field Programmable Gate Array (FPGA) which gets a playback program by the host computer containing all the configurations and instructions, and executes it on the chip. The recorded data on the chip is then sent back to the FPGA and can be read out by the connected computer.

[1] Change 12143 with ID I32e579... and Change 12424 with ID I26453b...

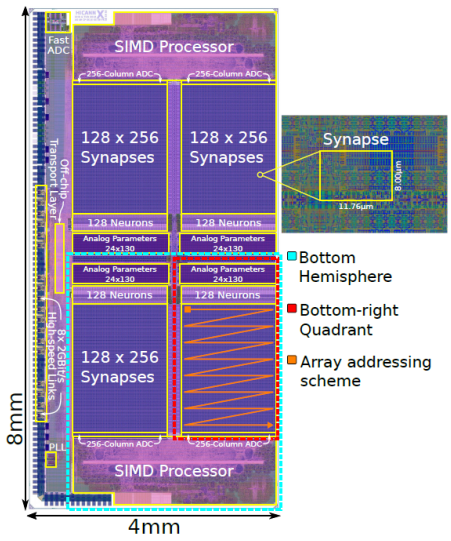

Figure 1: Schematic setup of the BrainScaleS-2 chip [3]

2.2 Software of BrainScaleS-2

In order to operate on neuromorphic hardware it is necessary to have software that translates the user's instructions into commands that can be executed by the chip. In my case, all the characterizations are made with the PyNN API which is based on Python and specialised for simulations on neural networks. It is based on other software stacks (haldls, stadls) which have a lower level of abstraction. In contrast to the others APIs in use, the focus of PyNN lies in the non-expert usability due to its high level abstraction of the hardware setup. It also allows to group several analog neuron circuits together in order to have one logical neuron. Currently the set values for the voltages, currents etc. must be entered in units of the digital-to-analog converter (DAC). It is planned to also give the possibility to set those parameters in biological values. The advantage of switching away from DAC values is that users don't have to apply a transformation themselves and have a better intuition for the values right away which leads to an improvement in the overall user friendliness. For more details on the implementation of PyNN for BSS-2 the according internship report should be read [2].

3 Characterization

In order to conduct the calibration of g_{leak} in the next section, it is necessary to investigate the behavior of V_{leak}, V_{th} and V_{reset} first. The characterization differs for each measurement that is taken. Not only does it differ for neurons

on different chips, but also for neurons on the same chip (cf. fixed pattern noise from chapter 2.1) and even for the same neuron for repeatedly conducted measurements (trial-to-trial variation).
In general, characterizations for the membrane voltage have already been done, however the aim of the internship was to conduct them on the default working point and examine how well PyNN is suited for such tasks. Important to mention is that a characterization of the Membrane Analog-to-Digital Converter (MADC) is necessary in the first place. Accordingly, a prewritten script was used for the following measurements and the experiments were conducted on the same chip (setup 68) to get less systematic errors.

3.1 Observing the membrane voltage

The first part of the advanced lab course investigates the firing behavior of a single neuron without input. Therefore, the neuron is brought into a continuous firing regime by setting V_{leak} above V_{th} (cf. figure 2a). A first comparison of different neurons is done in task 2a by recording the membrane voltage of four neurons with the same parameters and plotting them together (cf. figure 2b).

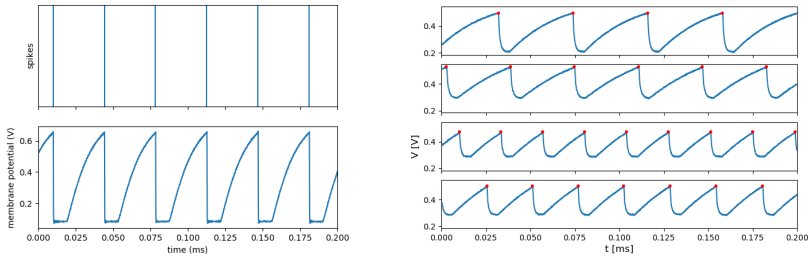

Figure 2: The membrane voltage in the continuous firing regime for a single neuron (left) and for four different neurons (right). The time when the neurons spike are indicated by the vertical lines (top left) and the red dotes (right).

3.2 Fixated DAC value for V_{leak}

For measuring V_{leak}, the neuron is brought into a stable regime, i.e. V_{th} is well above V_{leak} to prevent any spiking (here: $V_{leak} = 800$), while the remaining values stay at their defaults.
Then the membrane voltage has been measured for 0.2 ms and the mean is associated with the neuron's leakage voltage. The 128 neurons that are selected are all in the same block to find the intra-block-deviation. The error between the different neurons has been observed and shows deviations around 5 % (first row in table 1). For the differences between the different blocks of one chip the previous step has been repeated for all 128 neurons of the respective block. The

obtained values for V_{leak} (cf. table 1 and figure 3) indicate that the different blocks are responsible for significantly different values between neurons (at least at the default working point) which partly deviate more than 20 % (between block 0 and block 2). Therefore, the following experiments are all conducted on the same block.

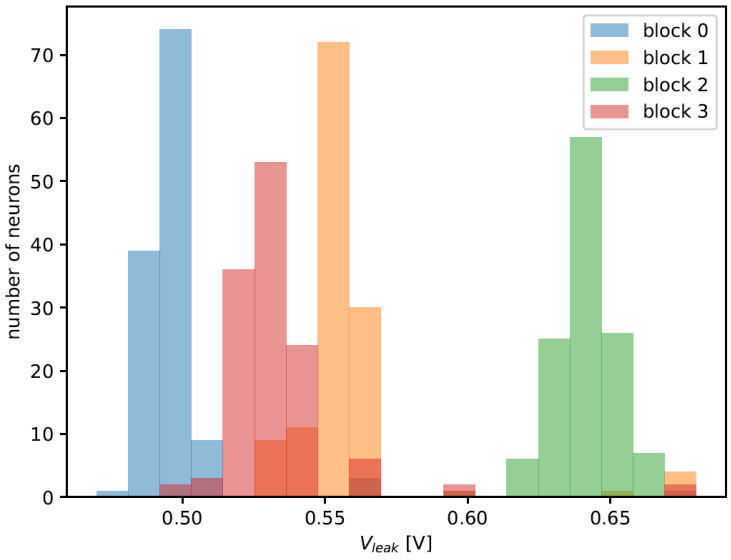

Figure 3: The histograms which contain the V_{leak} values of all 128 neurons of each block show significant deviations

Block	$V_{leak}[mV]$	Error in %
0	499 ± 22	4.41
1	557 ± 25	4.49
2	644 ± 15	2.33
3	534 ± 22	4.12

Table 1: Differences in V_{leak} due to the differences in the blocks

3.3 Whole DAC value range for V_{leak}

To set the histogram into context, it is necessary to examine the dynamic range of the neurons. Therefore, figure 4 shows the values of 30 neurons for V_{leak} over all possible DAC values (gray lines). The limitation to 30 neurons instead of all 128 neurons of one block is due to time constrains for the experiment duration. It can be seen that the linear regime starts when the DAC value is

set to approximately 500. The measured values for V_{leak} then starts at $0.073 \pm 0.020\,V$ and ends at $0.78 \pm 0.16\,V$. With a fit in the linear regime from a x-value of 540 until the last measurement at 1020, one obtains the translation $y = 0.001269 \times x - 0.514$ to determine the actual voltage (y-value) out of the set DAC value (x-value).

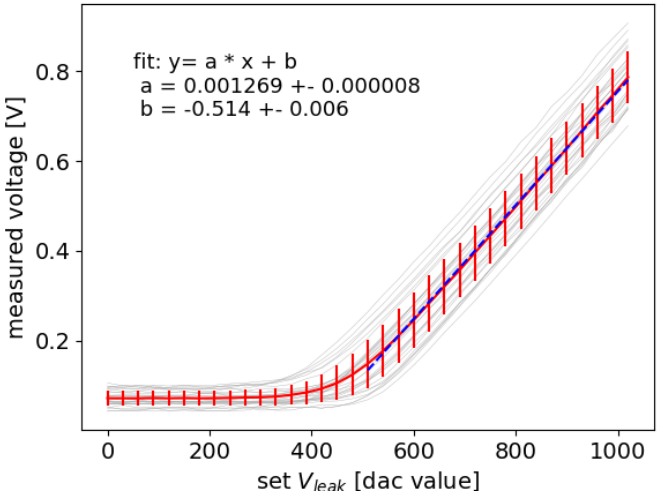

Figure 4: The gray lines show the V_{leak} values for 30 different neurons over the whole DAC value range, the red line is the mean of the single runs and used for the linear fit (blue)

3.4 V_{th} and V_{reset}

In order to bring the neuron into a continuous spiking regime, the value for V_{th} is set well below V_{leak} (here: $V_{th} = 400$). Whereas, V_{leak} has been set to its maximum of 1022 for the purpose of observing a spiking chip for the largest V_{th} range possible. With this setup, the maximum value of each action potential is associated with V_{th} while the minimum value belongs to V_{reset}. In a similar manner as with V_{leak}, the measured voltages for the whole range of the two parameters are observed.
Figure 5 shows on the left how V_{th} changes for increasing DAC values. For values below 0.8 V (or 550 in DAC values) one can observe a linear increase which behaves according to $y = 0.001432 \times x + 0.050$ (the fit is from 90 to 510). The plateau occures because the chip behavior changes from spiking to non-spiking as soon as V_{th} crosses the value of V_{leak}. Since the latter is set to 1022, we can calculate a leakage voltage of 0.783 V with the linear fit from figure 4. This is in good accordance with the plateau at $0.802 \pm 0.053\,V$.
In figure 5 on the right side, one can see how V_{reset} behaves for increasing DAC

values. As expected, this graph shows nearly the same behavior as V_{leak} (cf. figure 4), therefore the linear fit (from 550 to 1020) yields almost the same translation, namely $y = 0.001246 \times x - 0.499$.

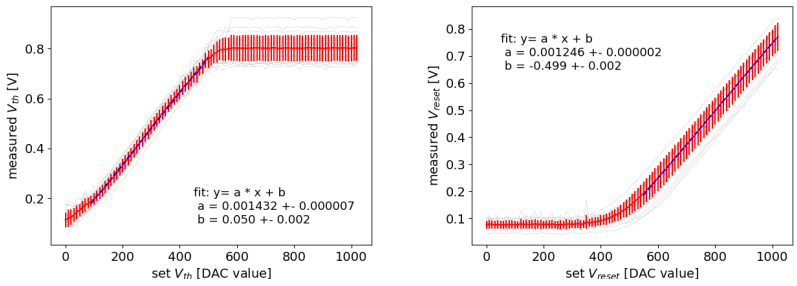

Figure 5: V_{th} (left) and V_{reset} (right) for 30 different neurons over the whole DAC value range, containing single runs (gray), mean (red) and linear fit (blue)

3.5 Chip-to-chip variation

The setups yield significantly different MADC characterizations in the first place (cf. figure 6) which can accounted for with the script mentioned in chapter 3. After doing so, the objective is to observe the differences in the characterization of V_{leak} (cf. chapter 3.3) in order to judge whether or not there is significant inter-chip variation that for instance can be caused by differences in the supply voltage.
As it can be seen from figure 7, there is a deviation for different setups. While the shape of setup 62 resembles the one of setup 68 (cf. figure 4) and only the translation differs, for setup 69 one can also see how the leakage voltage goes into saturation for DAC values at the upper end of the range.

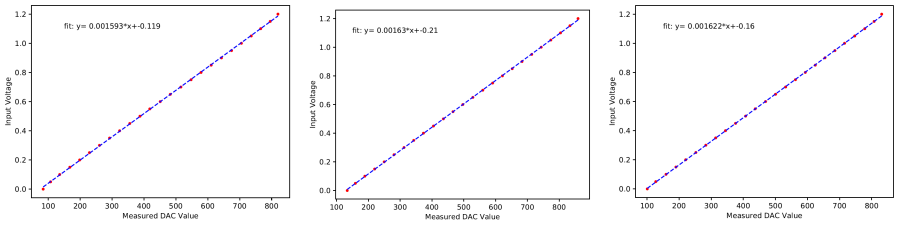

Figure 6: MADC characterization for setup 62 (left), setup 68 (middle) and setup 69 (right) with the respective linear translation fit (blue)

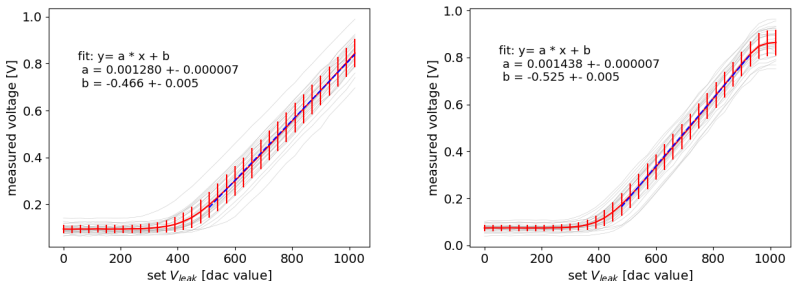

Figure 7: V_{leak} characterization for setup 62 (left) and setup 69 (right)

3.6 Trial-to-trail variation

For the in chapter 4 following calibration of the leak conductance based on the interspike intervall, it will be reasonable to characterize the trial-to-trial variation of g_{leak} first. Therefore, the aim was to measure the leak conductance 500 times with exactly the same parameters for the same neuron. The result (cf. figure 8) is a leak conductance of $g_{leak} = (3.30 \pm 0.07) \times 10^{-10}\,S$ which means a trial-to-trial variation of 2.1 % for this specific neuron and therefore lies in the expected range for deviations caused by noise (here coming from the power supply and digital crosstalk). However, the variation is probably in large parts due to the method of measurement since related quantities have already been determined more precisely.

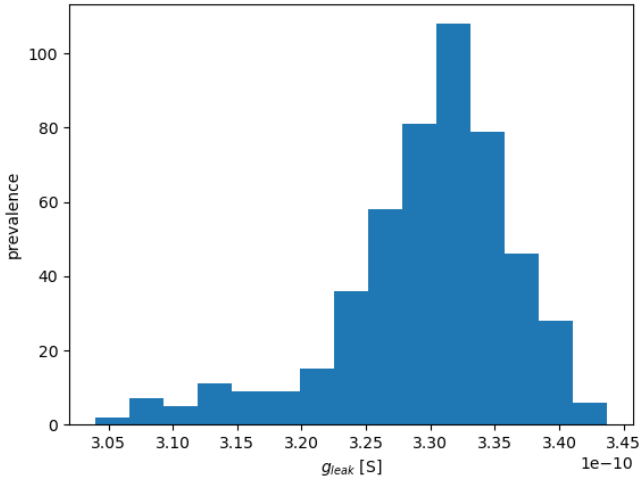

Figure 8: Histogram containing 500 g_{leak} measurements for one neuron with constant parameters

4 Calibration of g_{leak}

To get the same result no matter what neuron on a chip is used for a measurement, it is important to conduct a calibration. That's why the second task in the advanced lab course aims to calibrate the leak conductance g_{leak} towards one chosen value.

4.1 Theoretical background

We start with Kirchhoff's equation for a standard leaky integrate and fire (LIF) neuron without synaptic input[1]:

$$C_m \dot{V}_m = I_C(t) = g_{leak}(V_{leak} - V_m) \tag{2}$$

This ODE has the solution:

$$V(t) = V_{leak} + (V_{reset} - V_{leak}) \cdot exp(-\frac{g_{leak}}{C_m} \cdot t) \tag{3}$$

To obtain the time it takes for the membrane voltage to rise from V_{reset} to V_{th}, it has to be:

$$V(t_{rise}) - V(t = 0) = V_{th} - V_{reset} \tag{4}$$

If V_{th} is set to the following, the rise time is simply $t_{rise} = C_m/g_{leak} = \tau_m$:

$$V_{th} = V_{leak} - 1/e \cdot (V_{leak} - V_{reset}) \tag{5}$$

Then the measured interspike intervall (ISI) is simply given by (where τ_{refrac} is the time it takes for the membrane voltage to go from V_{th} to V_{reset}):

$$ISI = \tau_m + \tau_{refrac} \tag{6}$$

In order to calculate $g_l = C_m/\tau_m$ we have to know τ_{refrac} which is given by the frequency of the internal clock (default: $f_{clock} = 10\,MHz$):

$$\tau_{ref} = \tau_{ref\_DAC\_value}/f_{clock} \tag{7}$$

For this calibration the membrane capacity has been set to the maximal DAC value of 63 which equals a capacity of approximately 2.2 pF.

4.2 Results

Using this theory, the ISI for four different neurons have been calibrated with a *proportional control* which operates according to the following formula:

$$g_{set\_new} = (\alpha \cdot \frac{g_{aim} - g_{measured}}{g_{measured}} + 1) \cdot g_{set\_old} \qquad 0 < \alpha \leq 1 \tag{8}$$

This yields a correction that is proportional to the difference between the aimed and the measured value of the leak conductance. The proportionality factor α

is set to 1 for the first measurement. This means that the set value in PyNN is increased by 10 % if the measured g_{leak} is 10 % lower than the desired value. The result for 50 iterations can be seen in figure 10a. It becomes apparent that this approach corrects too much for the deviations and results in an oscillating behavior of the g_{leak} values. As a consequence, the first step was to test whether a linear change in the set value of the bias current which is used to adjust g_{leak}, really results in a linear change for the measured g_{leak}. Therefore, it was necessary to characterize the measured g_{leak} for the whole range of values that can be set for the DAC. The result (figure 9) is a clearly linear and therefore it is not the cause for the oscillating behavior.

With this verification, the next step was to vary the parameter α between 0

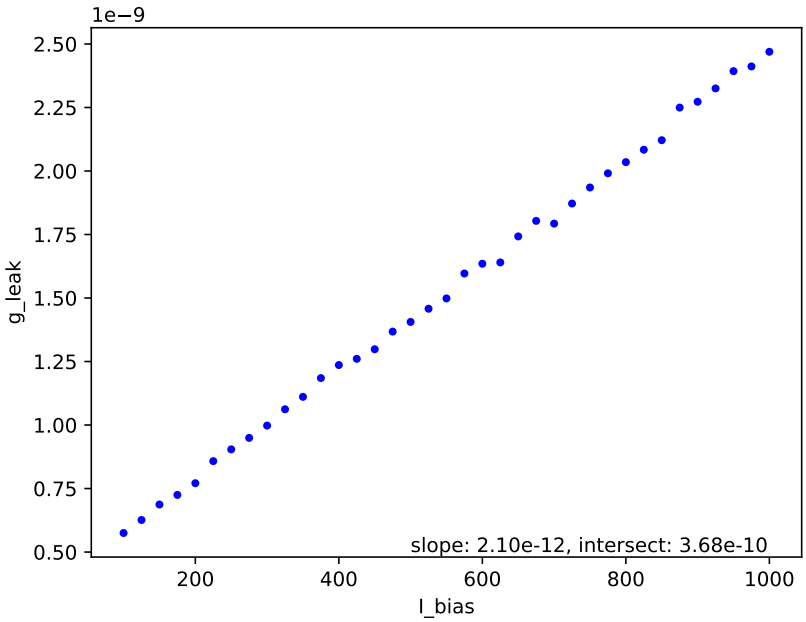

Figure 9: Measured g_{leak} over the whole DAC value range with parameters of the linear fit

and 1 which causes a weaker calibration than before. The results for different values for this parameter can be seen in figure 10. From this heuristic approach one could estimate that a value of around $\alpha = 0.25$ leads to the best calibration. However, the calibration reaches its limit if the difference in g_{leak} for two adjacent iteration steps has the same magnitude as the trial-to-trial variation (cf. subsection 3.6).

For the lab course, it is more suitable to have less iterations which means a

shorter runtime for the experiment. Then the proportional parameter should be increased to get an effectual calibration. In figure 11 an example is made with two iterations, a target firing rate of 50 kHz and α set to 0.9.

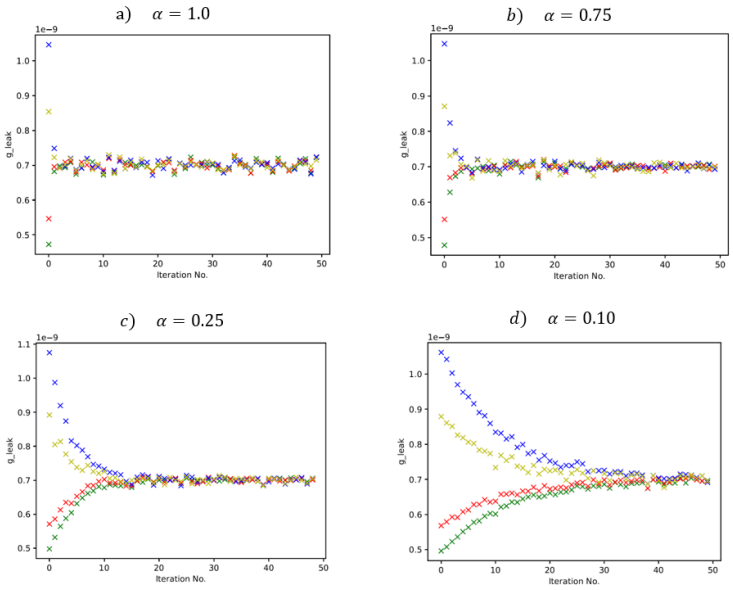

Figure 10: Each color represents one neuron that is calibrated over 50 iterations using different values for the calibration parameter α

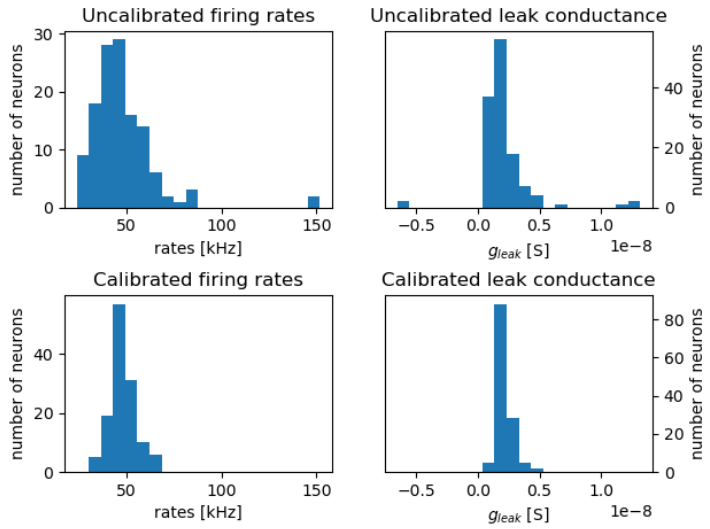

Figure 11: The uncalibrated setup (top) has a firing rate of 47 ± 18 kHz and a leak conductance of $(2.1 \pm 2.2) \times 10^{-9}$ S while the calibrated setup (bottom) has a value of 48 ± 7 kHz for the firing rate and $(2.1 \pm 0.7) \times 10^{-9}$ S for g_{leak}.

5 Summary & Outlook

The main task of the internship was to characterize the membrane voltage in order to find a translation from DAC values into voltages. With that in place, the first two tasks of the advanced lab course could be transported onto the new chip generation. Additionally, it was possible to examine and quantify the different variations. While the trial-to-trial variation could be determined around 2 %, the intra-block variation was a magnitude higher and also the deviation between different chips has to be accounted for. Consequently, the characterization of the membrane voltage has been all done on the same block of neurons. There, it could be shown that V_{leak} and V_{reset} effectively show the same behavior and within their errors yield the same (linear) translation from DAC values into voltage values. In contrast, V_{th} has its linear regime in the lower half of the DAC value range although it shares the same Volt range.
The calibration of the leak conductance has been successfully realised by adapting the proportionality parameter which is responsible for the strength of the correction. Naturally, the limitation of the calibration is given by the trial-to-trial variation.
The next steps involve the implementation of the found translation into PyNN and thereby accounting for the intra- and inter-chip variation. The idea is to create new types of neurons that can be selected in PyNN, so that it is possible to switch between calibrated and uncalibrated neurons as well as between neurons in hardware and biological parameters. However, the main task will be to

transfer the remaining scripts of the advanced lab onto the new setup and to amend it with new experiments that demonstrate the new features of the chip (this might involve exploring the capabilities of the PPU).

References

[1] Andreas Grübl; Andreas Baumbach. *F09/F10 Neuromorphic Computing*. 2006. URL: https://www.physi.uni-heidelberg.de/Einrichtungen/FP/anleitungen/F09.pdf.

[2] Milena Czierlinski. *PyNN Populations for BrainScaleS-2*. 2020. URL: https://www.kip.uni-heidelberg.de/vision/publications/reports/report_milenacz.pdf.

[3] Eric Müller; Christian Mauch; Philipp Spilger; Oliver Julien Breitwieser; Johann Klähn; David Stöckel; Timo Wunderlich; Johannes Schemmel. *Extending BrainScaleS OS for BrainScaleS-2*. 2020. URL: https://arxiv.org/abs/2003.13750.