Calibration of Synaptic Drivers on the Spikey Chip

Venelin Petkov

9. November 2012

Supervisors: Johannes Schemmel, Mihai Petrovici

1 Introduction

The core component of the FACETS neuromorphic hardware is a mixed-signal[1] ASIC[2], dubbed the Spikey chip, that is created using industrial photolithographic methods. This technology does not allow for the production of perfect copies of the building blocks of the circuits (transistors, capacitors, resistors, etc.), and therefore, the manufacturing process inevitably introduces variations in the electrical characteristics of these components. While this is of little or no consequence for the digital part of the chip, its analog circuits are highly influenced by such variations. Since all neuron and synapse circuits contain analog components, the parameters of the models they implement are never precise, but rather distributed around the target value.
models[3] that they implement are governed by normally distributed parameters[4].

The most significant advantage of the Spikey chip its configurability, allowing for the adjustment of most neuron and synaptic parameters. Process variations in the underlying circuits mean that when we set a particular target value for a neuron/synapse parameter, the value of its physical implementation on the chip will differ from the target. The relationship (transfer function) between the control variable and the process variable (in the lingo of control theory) is monotonic, but non-linear, and is uniquely parametrized for each unit - neuron, synaptic driver, etc. Control variables can take a relatively wide range of values by design, thus ensuring adequate provisions for the calibration of the chips.

The most challenging aspect of the calibration is the measurement of the process variable. While it is straightforward in the case of neuron potentials (reset,

[1] Contains analog and digital circuits
[2] Application-specific integrated circuit
[3] The hardware is a physical implementation of the LIF/AdEx model
[4] Process variations are expected to be Gaussian

threshold, resting), which can be directly accessed via oscilloscope, the measurement of other quantities such as the membrane time constant and the synaptic weights require the calculation of quantities derived from different measurements or the averaging of many experiment runs that may significantly slow down the process. Moreover, in the case of the latter, the measurements must be done in a particular network activity regime. This lab report presents methods and results for the calibration of the synaptic conductance courses that are generated by the Spikey chip.

2 System Description

The proposed calibration techniques have been tested on the Spikey chip, but have been implemented as a modular software framework that is expected to be adapted for use with HICANN[7] chips. A brief overview of the Spikey system is in order. More detailed information can be found in [1], [5], [4], and[6].

2.1 Implemented Neuron Model

Spikey v4 implements a leaky integrate-and-fire (LIF) model with conductance-based synapses. It also supports short-term plasticity (STP) and spike-time dependent plasticity (STDP), but those two capabilities will be omitted from now on since they are not necessary for the calibration methods discussed in this report.

The non-liner neuron model is described by the following differential equation:

$$-C_\mathrm{m}\frac{dV_\mathrm{m}(t)}{dt} = g_\mathrm{l}(V_\mathrm{m}(t)-E_\mathrm{l})+\sum_j p_j(t)(V_\mathrm{m}(t)-E_\mathrm{e})+\sum_k p_k(t)(V_\mathrm{m}(t)-E_\mathrm{i}) \quad (1)$$

When the membrane potential reaches a particular threshold V_thresh, it is pulled to the reset potential and remains at that potential for an absolute refractory time τ_ref:

$$V_\mathrm{m}(t) = \begin{cases} V_\mathrm{reset}(t), & \text{if } V_\mathrm{m} > V_\mathrm{thresh} \\ V_\mathrm{m}(t), & \text{otherwise} \end{cases} \quad (2)$$

The quantities involved in the preceding equation are summarized as follows:

- $V_\mathrm{m}(t)$ - Time-evolving neuron membrane potential
- C_m - Membrane capacitance
- g_l - Membrane leakage conductance
- E_l - Membrane leakage reversal potential or rest potential V_reset

- $p_{j,k}(t)$ - Conductance courses for the excitatory and inhibitory synapses
- $g_{j,k}(t)$ - Multiplicative factor describing the evolution of synaptic weights in time, due to synaptic plasticity
- E_e, E_i - Excitatory and inhibitory synaptic reversal potentials

Not all of the parameters entering equations 1 and 2 are configurable or adjustable on an individual neuron basis. Due to a trade-off that was necessary for more efficient die area usage, the neuron voltage potentials potentials V_rest, V_reset, and V_thresh, as well as the reversial potential E_i, are shared for a group of neurons (odd and even-indexed neurons on a neuron block, see[6] for details). The membrane capacitance C_m is fixed, since it is implemented by a physical capactor on the chip, while the membrane leakage conductance g_l can be adjusted individually for each neuron, allowing for the calibration of the membrance time constant $\tau_\mathrm{ref} = \frac{C_\mathrm{m}}{g_\mathrm{l}}$. The reversal potential E_e is also fixed at 0 mV BVD [5].

2.2 Conductance-Based Synapses

As we have seen in equation 1, the instantaneous membrane potential $V_\mathrm{m}(t)$ depends on the momentary conductance p?j, k(t) and the difference between itself and the corresponding reversal potential, which is the cause of the nonlinear behavior of the neuron. The hardware implementation of this part of the equation mimicks the biological process where ion channels at the synapse open and close with time, resulting in an alpha-shaped total conductance course. In lieu of having a number of discrete ion channels, however, the hardware simply changes the conductance between $V_\mathrm{m}(t) - E_\mathrm{rev}$ directly, thus achieving the same effect. The shape of this conductance course is closely related to the way it is generated by a multi-step process which we need to examine next.

Figure 1 depicts the structural unit of the Spikey chip (a neuron block and a synapse array) and sketches out the generation of the conductance course. The lower row of the block is comprised of 192 neurons(squares) that are connected to 256 synaptic drivers on the left(triangles).

When a spike is generated by a neuron or is fed as an external input into the network, it is represented by a digital pulse that enters a synaptic driver. This can be thought of as a start signal for the driver that elicits the generation of a voltage ramp, as shown in figure 2. Each driver sends this ramp to 192 synaptic nodes, one for each neuron. The synaptic node converts the voltage ramp into a current pulse that mirrors the shape of the desired conductance course (see figure 3) and scales it by a 4-bit weight. This current reaches an operational transconductance amplifier(OTA) that ultimately changes the conductance between the membrane potential and the reversal potential. In order to distinguish

[5] In order to differentiate between Voltages and Times in hardware and in the models (theoretical and software simulations), we have to specify the particular Domain to which they belong, resulting in HVD, BVD, HTD, and BTD, respectively.

between excitatory and inhibitory synapses, there are two separate lines for the control currents.

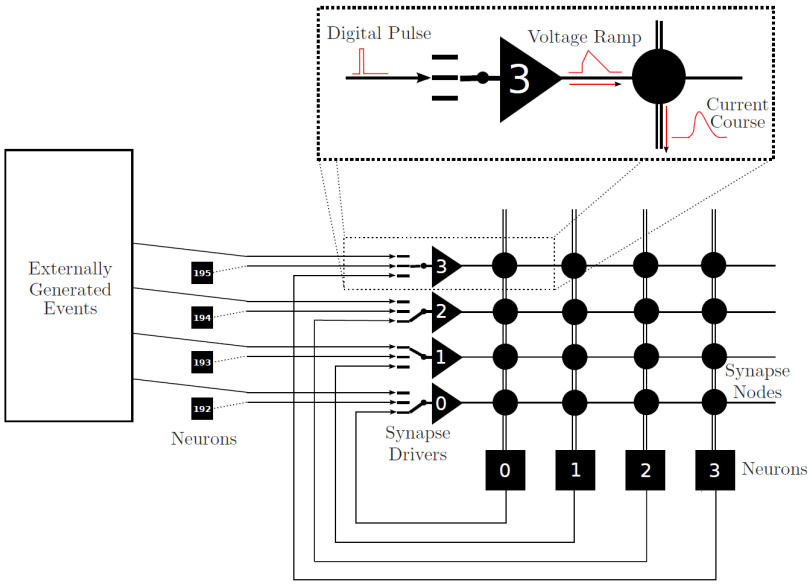

Abbildung 1: A schematic depiction of the synapse array. The inset depicts the generation of the conductance courses.

Due to their function, the synaptic drivers are necessarily analog devices and as such underlie process variations. In order to compensate for such effects the parameters that govern the generation of the voltage ramp are adjustable to allow for their calibration. The following table summarizes the corresponding configuration settings:

Parameter	Name	Configurability	Value Range
V_{start}	`v_start`	global	$[0, 1.8]$ V
I_{amp}	`drviout`	individual	$[0, 2.5]$ μA
I_{rise}	`drvirise`	individual	$[0, 2.5]$ μA
I_{fall}	`drvifall`	individual	$[0, 2.5]$ μA

Tabelle 1: Readout and input connection parameters

We can see from figure 3 that the slopes of the voltage ramp control the time constants of the conductance course while its height is directly proportional to its maximum.

Although the same voltage signal reaches all synaptic nodes, the conversion process relies on analog components too, so there is neuron-to-neuron variati-

on in the generated conductance course even if the voltage ramp is perfectly calibrated. There are no provisions for the calibration of this process and the only possibility remains a coarse calibration (due to the 4-bit weights) of the conductance course height by adjusting the weights for each neuron.

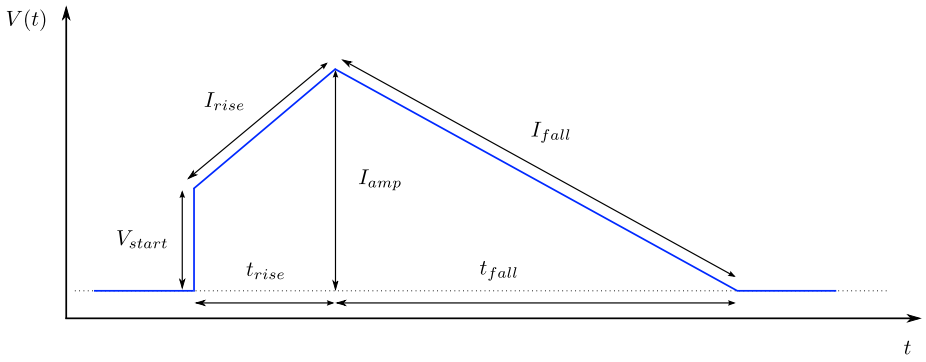

Abbildung 2: Voltage ramp generated by the synaptic driver

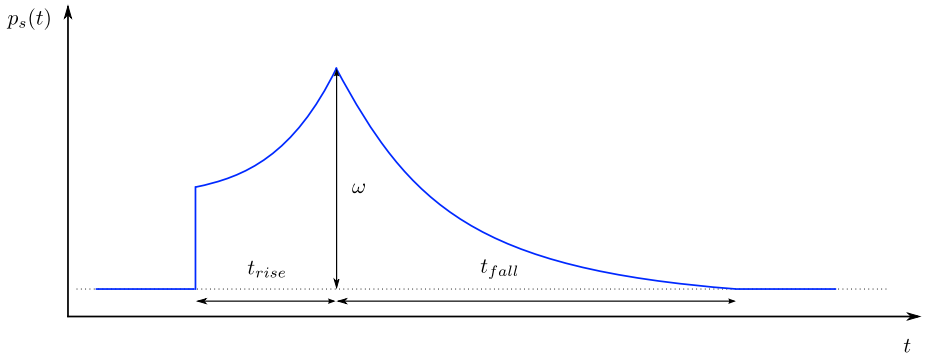

Abbildung 3: Conductance course

3 Calibration Methods

Due to the similarity in the controlling mechanisms used by the Spikey chip for most of its parameters, it is possible to develop a common calibration procedure where only the measurement routines are task-specific. It should also be noted that the calibration of synaptic drivers requires that the membrane time constant is set at the correct value, which implies that several neuron parameters have to be in the calibrated state already.

3.1 Binary Search

An important method that is used for most calibration tasks presented here is the binary search method[3]. Many different model parameters on the neuromorphic hardware are controlled by currents that are adjustable within the range of 0 - 2.5 μA. The transfer function for most process variables (membrane time constant, synaptic time constant, PSP height, etc) is in most cases monotonic but non-linear. Moreover, the measurement methods are often costly in terms of execution time, so an efficient method for finding the correct settings is needed. The binary search algorithm bisects the parameter space in two halves and then repeats the search in the partition where the target value of the process variable lies. This means that it is logarithmically efficient, thus executing only for a few steps (on the order of 10) before finding the desired setting. A small complication in its direct application to measurements on the hardware is that most quantities display a small trial-to-trial variability for a fixed control setting. Therefore, a certain tolerance (emipirically found) must be allowed for, in order for the method to function effectively (if it accidentially falls into the wrong partition, it will not terminate). On the practical side, a general version of the algorithm has been implemented within the calibration software framework, allowing for its application for many different purposes with adjustable tolerance, maximum number of iterations and control variable ranges.

3.2 Dynamic Range Calibration

The membrane potential voltage of the hardware neuron has a different range than the one expected biologically and must therefore be converted from HVD to BVD by a linear transformation. Nonetheless, the setting of particular potential values for V_{thresh}, V_{rest}, and V_{reset} is not done exactly, but has a small error ΔW associated with it, due to the use of operational amplifiers. It follows that the actual HVD values differ from the target HVD values by a fixed amount for each neuron. Moreover, the mechanism for reading out the membrane potential of each neuron also relies on an operation amplifier, thus adding an additional error ΔR for each measurement. Finally, there are four different lines that connect the neurons on one side of the chip to an external device, such as an oscilloscope, resulting in an error ΔL. We see that the expected uncertainty for setting a particular parameter is $\Delta\text{X} = \Delta\text{R} + \Delta\text{W} + \Delta\text{L}$. Fortunately, the relative error between the potentials is only ΔW. As the dynamics of the neuron do not depend on absolute values of the potentials, we are justified in defining the measured values as the ones we are setting, due to the constant $\Delta\text{R} + \Delta\text{L}$ for a particular neuron. Since most of the neuron dynamics is in the range V_{thresh} - V_{rest}, we can interpret this interval as the dynamic range and use the measured values to define a linear transformation HVD $\rightarrow$ BVD for each neuron. This is the first calibration must be performed with the software framework.

The rest potential V_{rest} is measured by averaging the rest potential of a neuron over several independent trials. The threshold and reset potentials are measu-

red by setting the threshold under the resting potential of the neuron. Figure 4 shows the corresponding HVD values for the whole neuron block. The horizontal lines represent the HVD targets (which can not be reached due to ΔX. For subsequent calibration purposes it is useful to pick a neuron whose potentials lie close to the measured means.

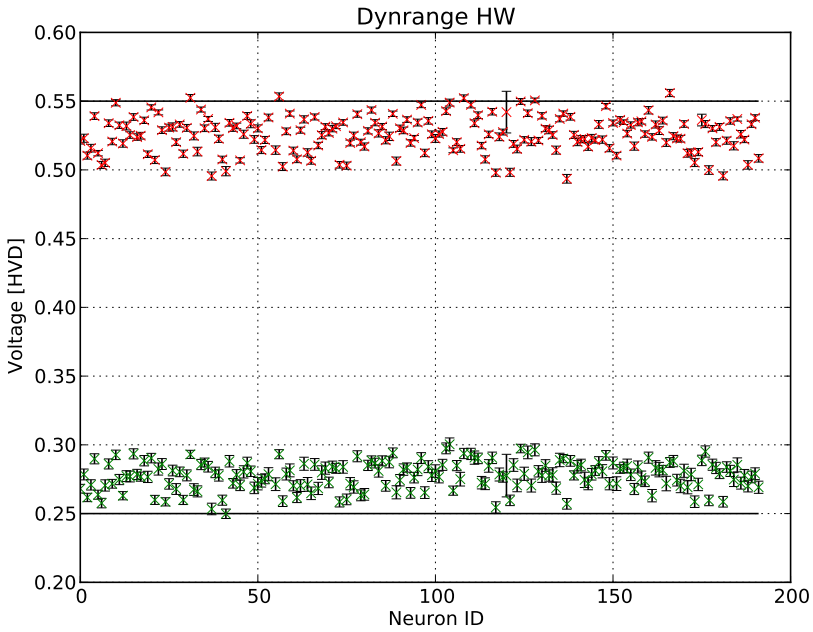

Abbildung 4: Measured threshold (red) and reset (green) potentials of all 192 neurons in a Spikey block.

The difference in the dynamic range of the neurons causes the same conductance course to have a different impact on the membrane potential. The only way to correct for this effect on Spikey is to find a multiplicative weight factor for each neuron to scale the PSP heights, the same solution proposed earlier for the homogenization of synaptic node variances. Due to the coarse resolution of 4 bits, such a calibration can mitigate the dynamic range differences only to a certain extend and also limits the available weight range.

3.3 Membrane Time Constant Calibration

Since the membrane time constant has a low-pass filtering effect on the conductance course, it is one of the factors that directly determine the shape of the PSP, from which we would like to infer conductance course parameters. In order

to achieve the correct activity regime for the simulation, τ_{m} must be properly calibrated prior to any PSP measurement.

3.3.1 Measurement

Due to process variations in the neuron potentials described above, the membrane time constant measurement requires the knowledge of the neuron-specific V_{thresh} and V_{reset} in advance. Along with interspike interval data, this allows for the calculation of τ_{m} as described in[6].

3.3.2 Calibration

The membrane time constants for each neuron are calibrated using the binary search method by adjusting the membrane leakage conductance. Figure 5 shows the result of this calibration on the FHW1v4 system, chip 445. The target value of 5 ms is the same for the measured τ_{m} in the uncalibrated state(blue) and the calibrated one(red) for a target value of 5 ms. The mean over all neurons in the first case is $\mu = 3.34$ ms with a standard deviation of $\sigma = 0.83$ ms, resulting in a coefficient of variation $\frac{\sigma}{\mu} = 0.25$ ms.

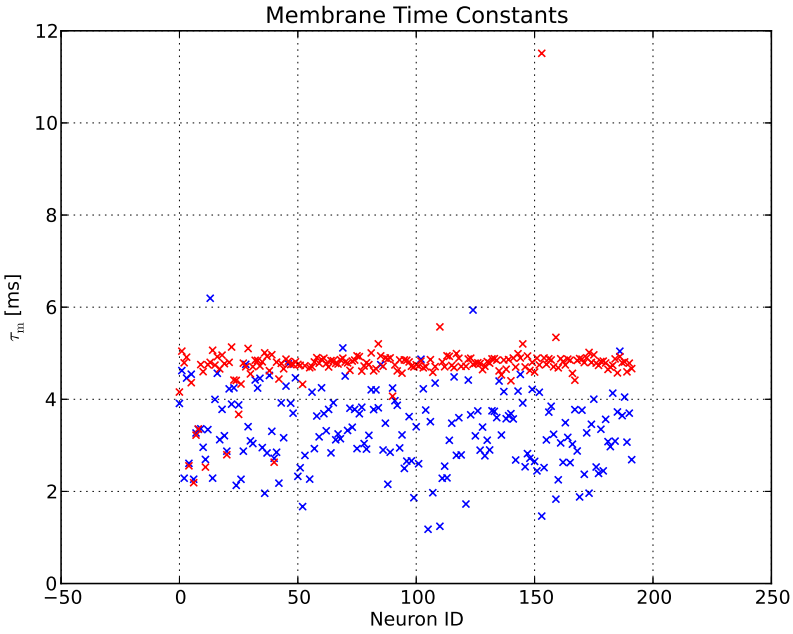

Abbildung 5: Calibrated (red) and uncalibrated(blue) membrane time constants.

The calibrated neurons, on the other hand, have a mean of $\mu = 4.73$ ms with a standard deviation of $\sigma = 0.66$ ms. The coefficient of variation displays a

marked decrease to $\frac{\sigma}{\mu} = 0.14$ ms and would obviously be even smaller if outliers (non-calibrated neurons) are excluded.

3.4 Synaptic Driver Calibration

Mixed-signal neuromorphic hardware does not strive to emulate software simulators of neural networks, but provides a novel approach to experimental neuroscience due to its speed and power efficiency. As we have seen, the parameters characterizing the implemented neuron/synapse models are in effect randomly distributed.
While this may be the case to some extent in biology, variation that can not be controlled may render the performed experiments meaningless (after all the implemented LIF/AdEx neurons are far simpler than biological neurons). It is therefore necessary to compensate such variations to a maximum possible extent.

The dynamics of the membrane potential is largely determined by the shape and timing of the post-synaptic potentials, and the former depend for the most part on the corresponding conductance courses. The greatest impact on the conductance course has the voltage ramp generated by the synapse driver. It is also the only aspect of conductance course generation that can be well-controlled on the Spikey chip. Therefore, the calibration of the synapse drivers is cruicial for most modeling tasks.

3.4.1 Measurement

FACETS/BrainScaleS neuromorphic hardware systems do not provide any facilities for measuring conductances directly. The parameters of a conductance course generated by a particular driver, therefore, have to be determined indirectly by measuring the resulting post-synaptic potential on a given neuron in a subthreshold regime.

High Conductance State As we have seen, the conductance between V_m and E_rev changes depending on the OTA control signal that is generated by a synaptic node and is propagated along a switch matrix column. This current lies in the microampere range, so that the parastic capacitance caused by the conductor material and the other elements on its path (all synaptic nodes share the same line to one neuron) have a dampening effect on the signal while they are charging. This means that when a single spike source is connected to a single neuron (conductance courses generated by a single synapse can not overlap in time on Spikey), it will produce a smaller, wider PSP than intended. This effect is difficult to correct for, since the current from each synaptic node must traverse a path of different length. In addition, this path contains slightly different capacitances for different neurons (switch matrix columns) due to process variations in the individual synaptic nodes. Finally, we do not know the absolute magnitude of the driver signal to begin with (again, due to process variations),

and process variations in the neuron circuit will result in a slightly different distance between V_{m} and E_{rev} for each neuron. Effectively, that means that the hardware can only be used for experiments where each neuron receives enough input to neutralize the parasitic capacitances and that the measurement of the post-synaptic potentials is valid only under these conditions.

Spike-Triggered Averaging The only way to measure the PSPs under realistic stimulation is to utilize the technique of spike-triggered averaging. When we are interested only in a single synaptic driver, the input generated by the rest can be seen as a background noise. With Poisson-distributed synapse contributions to this noise, the membrane potential of the neuron, $V_{\mathrm{m}}(t)$ is normally distributed in time. Overlayed on this background noise are the PSPs generated by the investigated driver. It is possible to determine the location of these PSPs from the spike times that signal conductance course generation in the driver. Taking the average of many small windows containing the PSPs reduces the variance of the background noise, thus revealing the shape of the mean PSP produced by the driver. In order to simplify the process, the driver can be programmed to fire periodically, so that the membrane potential trace can simply be sliced into equidistant windows that are easily averaged. This is the technique used in this lab for measuring the post-synaptic potentials.

Working Point The measurement of the parameters that govern a particular conductance course can be determined from the resulting PSP by fitting a function that theoretically describes it. As the experiments on the chip must occur in higher conductivity regimes though, we are faced with several difficulties when trying to implement this measurement:

- PSPs are dependent on the total conductance and the corresponding conductance course parameters that are used as target values must therefore be calculated in advance.
- Individual conductance courses on the Spikey chip are randomly parameterzed due to process variations, which makes it impossible to predict the total conductance.
- In contrast to theoretical models, the synaptic time constant on Spikey is also conductance-dependent, therefore the calibration can only be done for a specific conductance regime.

The first problem can be addressed relatively easy by performing STA in a software simulation under the desired conditions, which will provide us with the exact parametrization that the hardware has to achieve. These can also be calculated directly, based on the theory described in[2].

If the conductance courses are alpha shaped (a fairly good assumption for the neuromorphic hardware), it is possible to calculate the mean total conductance

by a single synapse with weight w, synaptic time constant τ_{syn}, and Poisson frequency ν in the following way:

$$g_{\text{syn}} := \langle g_{\text{syn}}(t) \rangle = \nu \; w \; \tau_{\text{syn}} \tag{3}$$

The mean total conductance is therefore the sum of all individual synapse contributions (excitatory or inhibitory) and the membrane leaking conductance:

$$g_{\text{tot}} := g_{\text{l}} + \sum_{\text{syn}} \nu_{\text{syn}} + w_{\text{syn}} + \tau_{\text{syn}} \tag{4}$$

Theoretically, setting all three parameters ν_{syn}, w_{syn}, and τ_{syn} to particular values will result in an effective potential (which can be equated with the mean membrane potential in this context):

$$V_{\text{eff}} = \sum_{\text{syn}} \frac{g_{\text{l}} + g_{\text{syn}} E_{\text{rev}}}{g_{\text{tot}}} \tag{5}$$

where E_{rev} is excitatory/inhibitory, depending on the synapse.

Setting these in hardware, however, will result in a different V_{eff} due to process variations. Therefore, the only way to determine the total conductance reliably, is to set two of the parameters and then adjust the remaining one until the theoretically determined effective potential is reached. The best candidate for the task is the synaptic weight w_{syn}, since the rate ν_{syn} is well-determined (spike delivery is controlled by digital electronics and is quite exact), and the control over the synaptic time constant τ_{syn} is more limited. Thus, changes in the synaptic strength (or weight in the theoretical sense) has the greatest impact on the PSP shape and should be used for calibration purposes.

There are two different lines for the excitatory and the inhibitory signals from the synaptic nodes. This means that the effective potential receives two independent contributions (barring unforeseen parasitic effects), so that the required synaptic strenghts for the excitatory and the inhibitory synapses can be determined independently and when used together will result in the correct V_{eff}.
This suggests the following procedure for establishing a well-defined working point in hardware:

a. Calculate the excitatory $V_{\text{eff,exc}}$ for the given N_{exc}, τ_{exc}, ν_{exc}, w_{exc}.

b. Set these parameters in hardware and vary the control current $I_{\text{amp,exc}}$ until the desired $V_{\text{eff,exc}}$ is reached (thus effectively finding a particular mean weight).

c. Turn on the inhibitory synapses with parameters N_{inh}, τ_{inh}, ν_{inh}, w_{inh} and vary $I_{\text{amp,inh}}$ until the target V_{eff} is reached.

In this state we have ensured that the background stimulation produces the correct mean total conductance g_{tot} and we can proceed with the characterization of the PSPs.

PSP Shape The ultimate goal of the measurement procedure is to fit a function that describes the expected form of the PSP in order to find the parameters best fitting the data. Usually, this is is an alpha function that can be expressed by the scaled difference of two exponentials:

$$V_{\mathrm{psp}}(t) = \omega \left(e^{-\frac{t}{\tau_{\mathrm{eff}}}} - e^{-\frac{t}{\tau_{\mathrm{syn}}}} \right) \tag{6}$$

The effective time constant depends on the total conductance, $\tau_{\mathrm{eff}} = \frac{C_{\mathrm{m}}}{g_{\mathrm{tot}}}$.

In the high conductance state the scaling factor ω can be calculated using the approximations described in[2] as follows:

$$\omega = \frac{w_{\mathrm{syn}} \left(E_{\mathrm{rev}} - V_{\mathrm{eff}} \right) \tau_{\mathrm{g}}}{g_{\mathrm{tot}} \tau_{\mathrm{eff}}} \tag{7}$$

with

$$\tau_{\mathrm{g}} = \left(\frac{1}{\tau_{\mathrm{syn}}} - \frac{1}{\tau_{eff}} \right)^{-1} \tag{8}$$

Theoretically, this fit provides us with a great deal of information:

- We can determine directly ω, τ_{eff}, and τ_{syn}
- From τ_{eff} we can determine g_{tot}
- From ω we can determine the weight w_{syn}.

Unfortunately, this does not work so well in practice, due to way that the conductance course is generated on Spikey. As we have seen, the controlling current that changes the conductance via the OTA has been designed according to the shape given in figure 3. The actual conductance course in hardware, however, could be more complex[6], which means that the estimation for τ_{eff} given by the fit is not valid. Since the alpha function is a difference of exponentials, and as such very sensitive to small changes in the constant, this means that the measurement of τ_{syn} has a great uncertainty associated with it. In addition, the value of the reversal potential E_{rev} is also not exactly known, due to process variations in the neuron circuit, so that we can not estimate the weight from ω very well (it also includes the great uncertainty from the time constants in τ_{g}.

One way to solve this problem is to calculate the shape of the PSP for the expected conductance course in hardware. The resulting function is fairly complex and with many degrees of freedom, which means that the fitting procedure can not be automated very well and often fails.

If we take into account the high-conductance state approximations used in the derivation of the PSP shape in[2], we can solve the resulting differential equation

[6] Personal communication with J.Schemmel

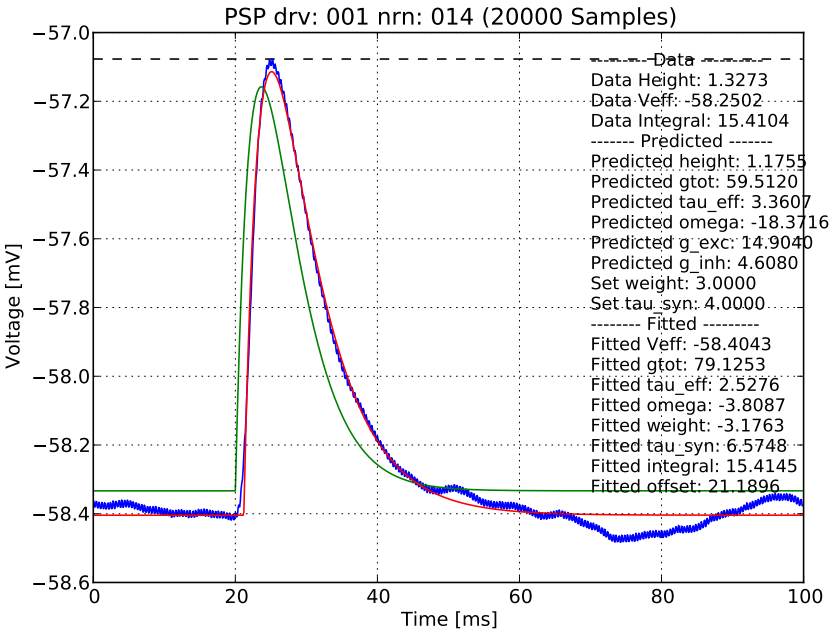

Abbildung 6: PSP with an alpha function fit

for a conductance course shown in figure 3 (under the simplifying assumption that the background conductance courses are alpha shaped). Unfortunately, the resulting function is fairly complex and difficult to automate the fitting process with, so it is better to assume a somewhat simpler conductance course where the instantaneous rising part (induced by V_{start}, see figure 2) is omitted.

The rising part of the conductance course is an exponential with a time constant λ that rises until time t_1, so that $\mathrm{w} = \exp(\mathrm{t}_1/\lambda) - 1$. The falling part of the curve is parameterized by the synaptic time constant, $\exp(-\mathrm{t}/\tau_{\text{syn}})$. By solving the equation for both region, we obtain the PSP shape:

$$V_{\mathrm{I}}(t) = A\tau_\mu \left(\mathrm{e}^{\mathrm{t}/\lambda} - \mathrm{e}^{-\mathrm{t}/\tau_{\text{eff}}}\right) + A\tau_{\text{eff}} \left(\mathrm{e}^{\mathrm{t}/\tau_{\text{eff}}} - 1\right) \text{ for } \mathrm{t} \in [t_0, t_1] \tag{9}$$

$$V_{\mathrm{II}}(t) = \omega \left(\mathrm{e}^{-\mathrm{t}/\tau_{\text{syn}}} - \mathrm{e}^{-\mathrm{t}/\tau_{\text{eff}}}\right) + \mathrm{V_I(t_1)}\mathrm{e}^{-\mathrm{t}/\tau_{\text{eff}}} \text{ for } \mathrm{t} > \mathrm{t}_1 \tag{10}$$

with

$$\tau_\mu = \left(\frac{1}{\tau_{\text{eff}}} + \frac{1}{\tau_\lambda}\right)^{-1} \tag{11}$$

$$\tau_{\mathrm{g}} = \left(\frac{1}{\tau_{\text{eff}}} - \frac{1}{\tau_{\text{syn}}}\right)^{-1} \tag{12}$$

$$A = \left(E_{\mathrm{rev}} - V_{\mathrm{eff}}\right)/C_{\mathrm{m}} \tag{13}$$

$$\omega = w\tau_{\mathrm{g}}\left(E_{\mathrm{rev}} - V_{\mathrm{eff}}\right)/C_{\mathrm{m}} \tag{14}$$

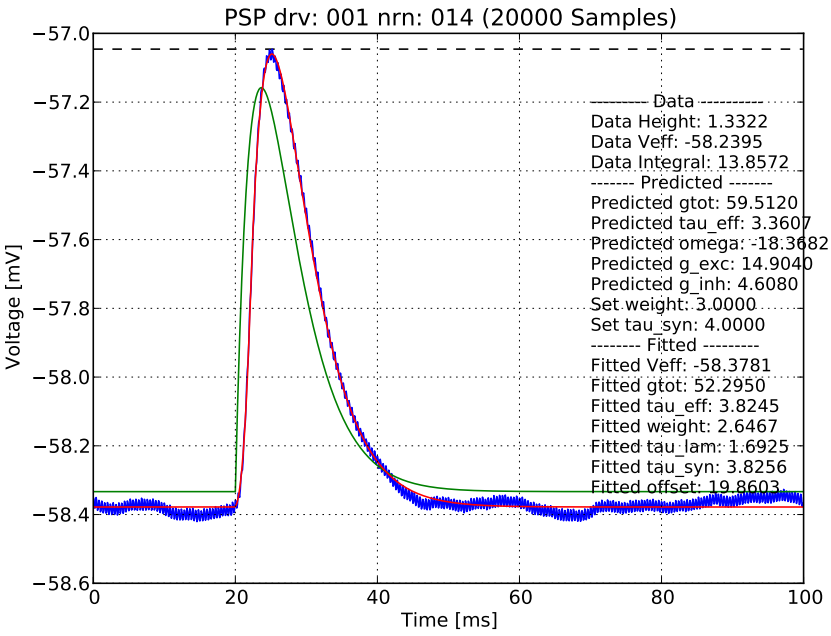

Abbildung 7: PSP fit with the custom conduction function solution

In any case, the alpha shape approximation is good enough for the hardware so that it should be used for calibration purposes, since there is no more appealing alternative.

3.4.2 Calibration

In order to perform the calibration of the synaptic drivers, the following steps are necessary:

Membrane Time Constant The calibration of synaptic drivers starts with the calibration of the membrane time constant of the neuron that will be used for this purpose. Based on the measurement of the dynamic range, the best choise for a neuron is one whose threshold and reset voltages are close to the mean values, measured for the whole neuron block. An optimal value that allows for the successful calibration of most neurons is $\tau_{\mathrm{m}} = 5$ ms.

Working Point Parameters Once this is done, the theoretical weights w_{syn} that are needed to reach the working point with given (by modeling considerations) ν_{syn} andτ_{syn} have to be found. This can be done either via software simulation with the WP method described above or approximately by calculating the needed weights.

Obtain Target PSP Parameters Using the working point parameters, perform STA in software or calculate the PSP theoretically in order to obtain target values for the calibration. At the moment the synaptic time constants can not be measured accurately (although the target value is known) from the PSPs, so the best way is to calibrate the amplitude of the PSPs. Due to the problems described above, measuring the weights from fits is somewhat less reliable than directly determining the PSP height from the data/theoretical calculations. Therefore, the calibration target for the synaptic driver calibration is the PSP height, defined as $|\max(V_{\text{PSP}}(t)) - V_{\text{eff}}|$.

Find the Working Point in Hardware In order to ensure that the PSP measurement is done in the proper conductance regime, the working point is found by a method parallel to the described software method. However, due to the nature of the hardware there are some important details. Firstly, all drivers produce PSPs of varying amplitude, so at the start of the WP calibration a particular set of excitatory and inhibitory drivers is chosen randomly from the available pool, excluding the driver that is to be calibrated. This configuration is retained throughout the WP and height calibration for this driver.

At the end of the PSP height calibration, drivers that achieve the target are marked as successfully calibrated. Their I_{amp} values are not changed during subsequent WP calibrations. Since this may cause a shift of the WP, the WP search is performed each time before a driver is calibrated.

The target weights obtained from the software simulation obviously do not correspond to the 4-bit weights that are used by the synaptic node to divide a PSP amplitude into 15 possible parts. There is an arbitrary translation factor that is used to discretize the theoretical weights and map them into the 0-15 range, but this is only a matter of convenience. Choosing a particular weight on the hardware is therefore independent from the actual weights, since the amplitude of the PSP can be adjusted by I_{amp}. When using a particular chip, however, it may be the case that the drivers are too weak or too strong to achieve the calibration target. Therefore a judicial setting for these bit-weights is necessary before performing the WP search.

PSP Height Calibration Using the target value for the PSP height, the binary search algorithm is used to find a setting for I_{amp} that will achieve that target within a particular tolerance. Drivers that can not be calibrated are left

uncalibrated. The number of successfully calibrated drivers may depend on the working point chosen for a particular experiment.

4 Calibration Results

The following calibration has been performed on the Spikey chip 445 using a LeCroy WR44Xi oscilloscope for the analog membrane trace measurements. Each single experiment on the chip had a length of 100000 ms BTD. A PSP window of 100 ms BTD therefore provides 1000 samples for averaging. Since good PSP measurements require at least 10000 samples, the oscilloscope is automatically configured by the measurement routine to average the membrane potential traces resulting from N experiments, saving the mean on a different channel which is read at the end of the measurement. In this way the data that has to be transferred from the scope is kept relatively small and the slicing of the resulting data array to obtain the mean PSP is very efficient. It has been determined that a number of N = 20000 samples is sufficient so that the variance of the τ_{syn} is kept within 1 ms (athough as we discussed above, the measurement itself is not necessarily accurate).

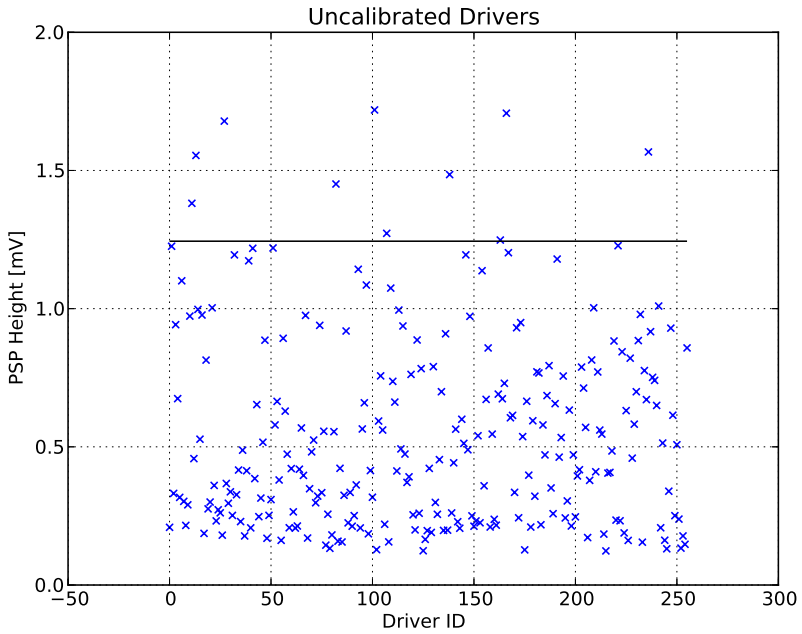

Abbildung 8: Uncalibrated drivers.

The working point has been adjusted before the calibration of each driver by

reassigning the background drivers randomly and subsequently performing the working point calibration as described above. Using the binary search algorithm, the control current I_{amp} has been adjusted until the PSP height matches the target with a tolerance of +/− 0.05 mV. The calibration of drivers that could not achieve half of the required height with 2/3 of the maximum I_{amp} has been aborted, so that they do not slow down the calibration process.

The bit-weights used for this calibration have been set to 6 for the excitatory and 4 for the inhibitory drivers. The calibration has been performed on neuron 14, whose membrane time constant has been calibrated for 5 ms. The target synaptic time constant has been set at 4 ms. The excitatory effective potential for the WP calibration was -55 mV, while the target effective potential has been set to -58.33 mV.

4.1 PSP Strength

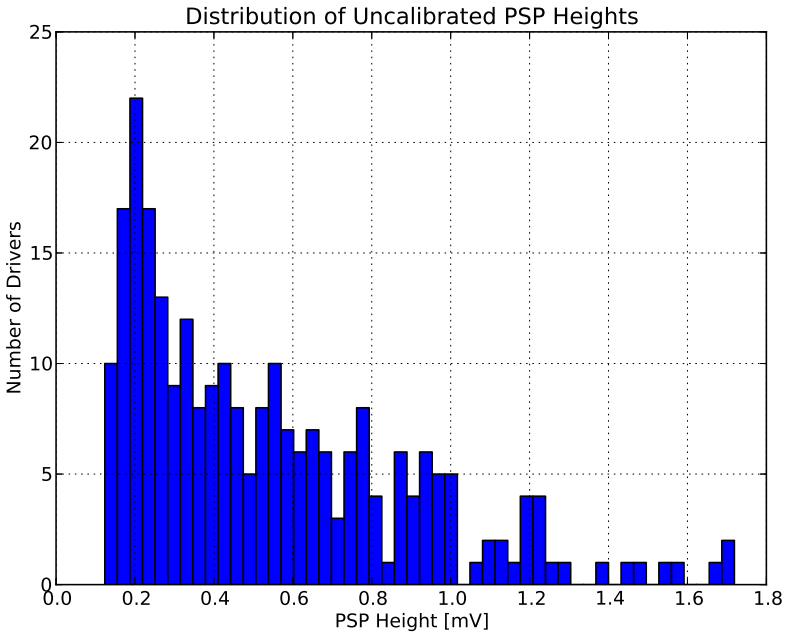

Abbildung 9: Distribution of uncalibrated drivers.

As it can be seen in figures 8 and 9, the uncalibrated drivers exhibit significant variation in their height. The horizontal line represents the target PSP height of 1.24 mV that has been determined by a software simulation. Before calibration the mean of the PSP heights is $\mu = 0.54$ mV, with a standard deviation

of $\sigma = 0.35$ mV. The coefficient of variation is therefore $\frac{\mu}{\sigma} = 0.65$ mV. For simulations in the high-conductance state where PSP amplitudes are generaly in the range of 1-2 mV, this variation will have significant impact.

Interestingly, the PSP heights seem to follow an exponential distribution, as seen on figure 9.

At the end of the calibration, 71 out of 256 excitatory drivers have been successfully adjusted to reach the target PSP height. This result is largely dependent on the characteristics of the particular chip and the required conductance regime. In this case a large number of drivers could not reach the target height even at maximum values of the control current I_{amp}. It is possible to optimize the yield empirically by increasing/decreasing the bit weights of the drivers, but this is a tradeoff against the weight configurability, because it would limit the range of the synaptic weights used by the experimenter when the bit weight is set too high or too low.

The bimodal distribution shown in figure 11 shows that some of the drivers can not be configured outside of a very narrow range, while others achieve successful calibration targets.

The mean PSP height of the successful drivers is $\mu = 1.21$ mV with a standard variation of $\sigma = 0.05$ mV, which is exactly the tolerance value used for the calibration. The small coefficient of variation, $\frac{\sigma}{\mu} = 0.04$ shows that drivers capable of achieving the target PSP height can be calibrated very accurately.

4.2 PSP Time Constant

Due to the considerations described above, the synaptic time constants τ_{syn} have not been calibrated and at the moment it is unclear how well it is possible for them to be calibrated. In the course of the lab, several approaches have been tried, but no consistent procedure was found that is comparable to the high accuracy of PSP height calibration. The problems are several:

- The PSP fitting method does not deliver reliable results, as already discussed
- The adjustment of I_{fall} that changes the falling slope of the voltage ramp used to generate the conductance course leads to massive changes in the height of the PSP (so possible τ_{syn} calibration must be done before the height calibration), which in turns causes the rise time of the PSP to change. The fitted τ_{eff} therefore varies for different settings of the synaptic time constant, leading to non-monotonous behavior that precludes calibration by the binary search method.
- The synaptic time constant seem to be adjustable in a very narrow range, leading to small possible yields.

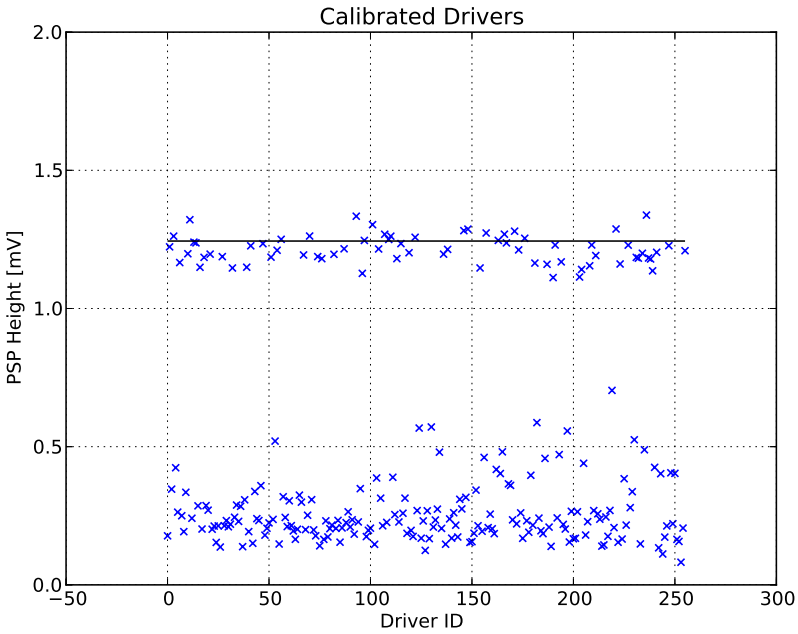

Abbildung 10: Calibrated drivers. The horizontal line represents the target height.

Despite these problems, as it can be seen on figure 12, there is some systematic deviation from the target value of 4 ms, since the mean is $\mu = 5.71$ ms, with a standard deviation of $\sigma = 3.07$ ms. The coefficient of variation, $\frac{\sigma}{\mu}$, is comparable to that of the uncalibrated PSP heights.

5 Calibration Framework

As we have seen, the calibration of the synaptic drivers requires a number of different measurement and calibration steps to be performed. This process was made possible by the development of a robust, failure-tolerant software framework that can be easily configured for use with different FHW1v4 chips in a way that automates as many functions as possible. This section describes the design goals and the architecture of that framework. Additional information about its utilization and a user manual is contained in the accompanying documentation.

The framework has been developed in the Python programming language, since it requires heavy use of PyNN and also requires some additional modules to access the chip and the hardware.

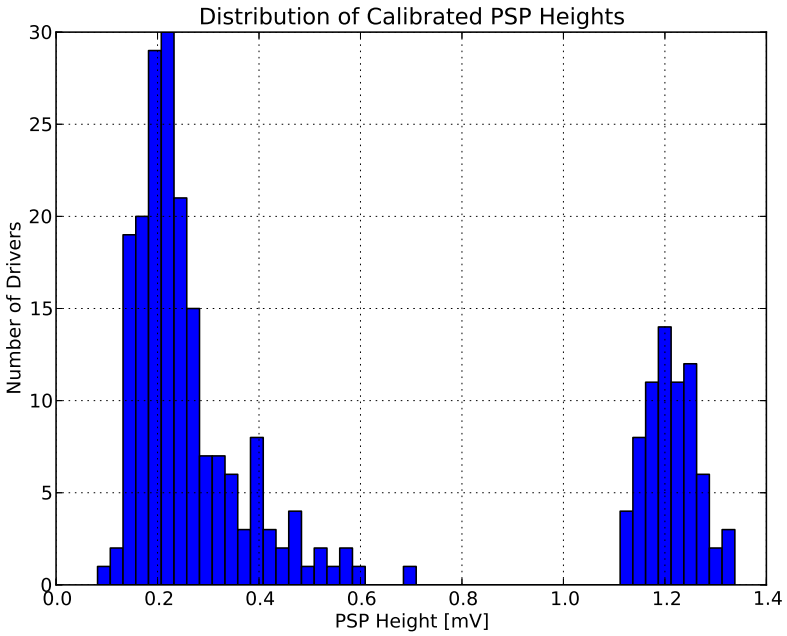

Abbildung 11: Distribution of calibrated drivers.

5.1 Design Goals

There are several design goals that the implementation of the framework strives to achieve:

- User-friendliness: The measurement and configuration commands are accessible through a standards-complying command-line interface, based on the argparse Python module. They are grouped in 3 main categories - measurement, calibration, and plotting. Since there are various dependencies (e.g. some calibration tasks can not be made before others are completed), an accounting process that is under development will remind the user of dependencies and prevent possibly erroneous operations.

- Automation: The measurements and calibration steps use a number of components and devices to accomplish their task. The user can not be expected to know all details about the low-level functionality, therefore the individual calibration steps are automated as much as possible. Complete automation, however, has proved illusive at the moment, since differences between hardware systems may require device-specific configuration for some of the calibration steps. Moreover, the whole process uses a number of different devices (oscilloscope, chips, etc.) that may need to be prepared prior to performing the calibration. Future neuromorphic hardware is

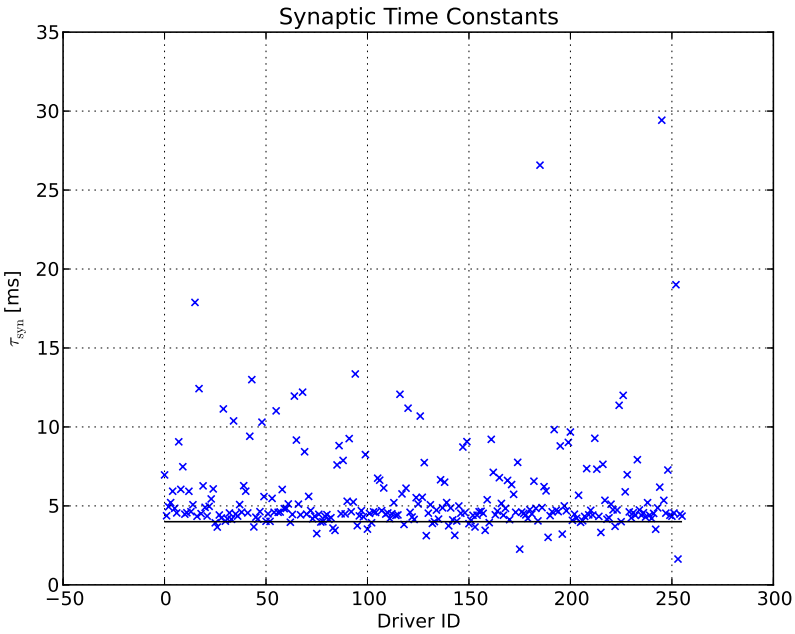

Abbildung 12: Synaptic time constants

expected to provide more integrated interfaces to access the experimental data that would enable complete automation.

- Modularity: All routines and algorithms that are reused for different measurements are encapsulated in functions or classes that can be used accross the framework. The measurement and calibration methods should be applicable with minimum reprogramming effort for future hardware versions.

- Backward compatibility: The framework is stand-alone so that it does not reqiure any legacy calibration code, but since it has been tested with Spikey, it produces calibration data that are recognized by the FHW1v4 system. Due to its modularity, it is expect that it will be easily extended to work with other formats and storage schemes.

- Robustness: Mixed-signal neuromorphic devices are designed to function even when some of their components are malfunctioning, so that a certain failure rate is expected. The calibration framework must deal with individual neurons and synaptic drivers that can not be calibrated. It should fail gracefully and notify the user. Since many measurements and calibration methods take a lot of time, they have been programmed in such a way that each action can be performed on individual neurons and synapse drivers.

This prevents the user from having to repeat hours-long calibration runs that terminate due to problems with individual components.

5.2 Architecture

In order to achieve the design goals outlined above, the framework has been developed around its command-line interfaces. A top-level module contains the definitions and the configuration options for the individual commands. These are hierarchical and organized in groups as already mentioned. For each command there is a Python function that is executed with the command-line arguments and options as its arguments. In addition to parsing the commands, the top-level module also loads the configuration file settings at startup and converts it to a Param object of the type described below. The last important function that is performed at startup is the instantiated of a Python logger object that can be used by all modules.

Each type of measurement is encapsulated by a special Meter class that can be reused by other parts of the framework, e.g. for calibration purposes.

Most measurements require to perform experiments on the hardware. They are largely stereotypical, using a single neuron with N excitatory and M inhibitory inputs (where N,M could be zero). In order to avoid writing different PyNN scripts for each type of measurement, a special class that covers all usage scenarios needed for the calibration has been developed. A great contribution to its flexibility is offered by the Parameter object.

A neuroscientific model typically depends on a large number of parameters. The ability to organize these parameters in a meaningful way and to manipulate them in order to change the behavior of the model is central for the measurement and calibration purposes described in this lab report. A clean hierarchical model using the dot notation (e.g. parameters.neuron.threshold, etc.) enables the user to group parameters in a very transparent way which simplifies the programming and utilization of the framework. The serialization and deserialization of the parameter object from plain-text files also simplifies debugging and simulation accounting.

The clear separation between individual measurement and calibration steps opens the possibility to plug in different routines and retain the same command-line interface for future neuromorphic devices. Currently all data regarding measured quantities, calibration settings, and simulation results are saved in plain-text files and numpy arrays. The calibration of the synaptic drivers of future neuromorphic devices may require a database for larger volumes of data.

5.3 Usage Manual

The user manual for the calibration framework can be found in its code repository at:

git@gitviz.kip.uni-heidelberg.de:spikey-calib.git

Please note that the code is still heavily under development and has not yet reached production-grade maturity. Until that milestone is achieved, please consult its documentation and developer.

6 Conclusion

Experiments on mixed-signal neuromorphic hardware differ significantly from software simulations due to the variability in neuron and synaptic parameters that is introduced by the manufacturing process. In the uncalibrated state such variations are large enough to preclude the emulation of many neuroscientific models. Since the largest contribution the behavior of the network is given by the post-synaptic potentials, it is crucial that the conductance courses generated by the hardware be calibrated. Based on [1], [2], and [6], we have developed calibration methods and a software framework for calibration that has been tested with the FHW1v4 system and can also be extended for usage with other neuromorphic hardware developed within the VISION(s) group. We have shown the distributions of PSP parameters in the calibrated and uncalibrated state, thus demonstrating the effectiveness of the method.

At this time the software has been used mainly with chip 445 and must be tested with other systems. In addition, while the main functionality is already implemented, several steps have to be completed, if the calibration process is to be automated further and applied to a whole block of neurons that can be used for network simulations.

Literatur

[1] Daniel Bruederle. *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System*. PhD thesis, University of Heidelberg, 2009.

[2] Ilja Bytchok. From shared input to correlated neuron dynamics: Development of a predictive framework. Master's thesis, University of Heidelberg, 2011.

[3] D.E.Knuth. *The Art of Computer Programming*. Addison-Wesley, 1997.

[4] Andreas Gruebl. *VLSI Implementation of a Spiking Neural Network*. PhD thesis, University of Heidelberg, 2007.

[5] Andreas Hartel. Improving and testing a mixed-signal vlsi neural network chip. Master's thesis, University of Heidelberg, 2010.

[6] Venelin Petkov. Toward belief propagation on neuromorphic hardware. Master's thesis, University of Heidelberg, 2012.

[7] J. Schemmel, J. Fieres, and K. Meier. Wafer-scale integration of analog neural networks. *Proceedings IJCNN2008*, 2008.