

**Department of Physics and Astronomy
University of Heidelberg**

Bachelor Thesis in Physics
submitted by

Anne-Christine Scherzer
born in Herrenberg (Germany)

2013

Phase-Locking on Neuromorphic Hardware

This Bachelor Thesis has been carried out by Anne-Christine Scherzer at the
Kirchhoff-Institute for Physics in Heidelberg
under the supervision of
Prof. Dr. Karlheinz Meier

Abstract

A barn owl's auditory system is remarkable because it can locate sounds with a very high azimuthal precision. This requires that the auditory neurons resolve interaural time differences of a magnitude smaller than the time constants of the involved neurons. In a publication of 1996, Gerstner and others presented a spiking neural network model which is capable of resolving these time differences. The key process leading to such a precision is termed *phase-locking*. Phase-locked neurons exhibit a very precise temporal spiking behavior, allowing even small time differences to be distinguished in further processing steps. The aim of this thesis is to show phase-locking on the Spikey chip. This requires several modifications of the originally proposed neuron and synapse models and their respective parameters, because neural networks to be emulated on this chip are restricted to its inherent neuron and synapse models as well as limited parameter ranges. Preliminary simulations with a hardware-inspired software model of the network confirm that phase-locking works well with the modified models and parameters. In the further course of this study, many of these parameters on hardware are measured and adjusted as well as possible to the parameters of the hardware-inspired software model. The adapted network is emulated on the Spikey chip and its performance is analyzed. It is shown that phase-locking can be achieved, which is an important step towards sound localization on neuromorphic hardware.

Kurzfassung

Schleiereulen besitzen einen bemerkenswerten Ortungssinn. Sie können Geräusche mit einer sehr hohen azimuthalen Genauigkeit lokalisieren, wofür kleine Zeitdifferenzen zwischen den Signalen von beiden Ohren anhand des Feuerverhaltens der beteiligten Neuronen detektiert werden müssen. Diese Neuronen besitzen Zeitkonstanten, die mindestens um eine Größenordnung größer sind als die Zeitdifferenzen, die aufgelöst werden sollen. In einer Veröffentlichung aus dem Jahr 1996 wurde von Gerstner und anderen ein neuronales Netzwerk vorgestellt, das in der Lage ist, solch kleine Zeitdifferenzen aufzulösen. Grundlegende Voraussetzung hierfür ist das sogenannte *Phase-Locking*. Neuronen mit Phase-Locking weisen ein sehr präzises zeitliches Feuerverhalten auf, das es ermöglicht, auch sehr geringe Zeitdifferenzen in weiteren Verarbeitungsschritten zu unterscheiden. Ziel dieser Arbeit ist es, Phase-Locking auf dem Spikey Chip zu erreichen. Hierfür sind einige Veränderungen der ursprünglichen Neuron- und Synapsenmodelle sowie der zugehörigen Parameter erforderlich, da die Neuronen- und Synapsenmodelle auf der Hardware vorgegeben sind und die entsprechenden Parameter nur in eingeschränkten Bereichen verändert werden können. Zunächst vorgenommene hardwarenahe Software-Simulationen des betrachteten Netzwerks bestätigen, dass Phase-Locking mit veränderten Parametern auf der Hardware erreicht werden kann. Im weiteren Verlauf dieser Arbeit werden daher viele dieser Parameter bestmöglich an die im hardwarenahen Software-Modell verwendeten Parameter angepasst. Mit dem so angepassten Netzwerk wird Phase-Locking auf dem Spikey Chip erreicht und untersucht. Damit kann ein wichtiger Schritt hin zu Geräuschortung mit neuromorpher Hardware vorgenommen werden.

Contents

1	Introduction	1
2	Network setup	3
2.1	Biological background	3
2.2	Presynaptic input	4
2.3	Learning process	6
2.4	Vector Strength	7
3	Hardware synapse model	9
3.1	Synaptic weight resolution	9
3.2	Weight update	9
3.3	Learning Window	10
4	Measurement and adaptation of hardware parameters	13
4.1	Presynaptic stimulus	13
4.2	Membrane time constant	13
4.3	Synaptic time constant	14
4.4	Learning Window	15
4.5	Weight recording	16
5	Results	17
5.1	Software Simulations	17
5.2	Hardware parameter modification	22
5.2.1	Membrane time constant	22
5.2.2	EPSP modification	24
5.2.3	STDP curves	26
5.3	Phase-locking on hardware	29
5.3.1	Learning results	29
5.3.2	Weight development	31
5.3.3	ITD dependency	34
5.3.4	Measurements without learning	36
6	Conclusion and outlook	39

1 Introduction

In a publication of 1996, Gerstner and others focused on the auditory system of the barn owl (Gerstner et al., 1996). A barn owl can locate sounds, and thus prey, with an azimuthal precision of 1-2 degrees. This requires the detection of very small phase differences in the range of a few microseconds between the signals from the left and the right ear. Such temporal precision is remarkable, because auditory neurons in the barn owl have been shown to have time constants which are at least one order of magnitude slower (Gerstner et al., 1996). Also, the signals from both ears may take different transmission lines to the nucleus laminaris, where sound localization takes place. The intrinsic variation in the delays of these transmission lines is very large compared to the time differences that need to be evaluated. Even if the neurons in each transmission line would spike very precisely, any phase information would still be lost on the way to the nucleus laminaris due to different delays. A combination of very precise spike timing of the auditory neurons and a selection of transmission lines with appropriate delays is therefore necessary to resolve such small phase differences. It is unlikely that the barn owl already possesses such accurately tuned transmission lines when it is born because its head grows considerably after birth (Maass and Bishop, 2001). Instead, there must be a way of training the auditory neurons to spike very precisely and of selecting the appropriate transmission lines that pass the input signals from both ears. The key process responsible for this is called *phase-locking*. A neuron is termed to emit phase-locked spikes when it fires preferably around a certain phase of a stimulating signal. In the case of the barn owl, auditory neurons are stimulated by a single frequency which they receive from the ear. In Gerstner et al. (1996), a network is introduced with which a neuron can be trained to show phase-locked spiking by means of a Hebbian learning rule. This learning rule selects synaptic delays from a broad Gaussian distribution of delays such that presynaptic spikes arrive coherently. If this is the case, a postsynaptic neuron receives input simultaneously from all presynaptic neurons at a certain phase of the input signal. This causes the postsynaptic neuron to emit spikes preferably around that phase. Consequently, it exhibits phase-locking.

During a preceding research internship (Scherzer, 2012), the neural network described in Gerstner et al. (1996) has been implemented with PyNN (PyNN, 2008) and simulated with NEST (NEST, 2008). This thesis focuses on transferring the network onto the Spikey chip, the aim being to train a postsynaptic neuron to show phase-locking as a first step towards sound localization on neuromorphic hardware. This involves the task of adapting the network to the hardware capabilities in several aspects. At the same time, the network's parameters are kept as close as possible to the original publication (Gerstner et al., 1996) to enable successful learning and for better comparison. A large part of this thesis will therefore deal with the measurement and optimization of those parameters which can not be set arbitrarily on hardware. This includes software simulations of the network with the modified parameters to ensure that phase-locking works with these parameters. With the network adapted to the hardware limitations, the learning process, which leads to phase-locking in a neuron's spiking behavior, will then be run on the Spikey chip and analyzed.

2 Network setup

The following sections will focus on the topology and properties of the neural network adapted from Gerstner et al. (1996). Presynaptic neurons are assumed to already exhibit periodic, phase-locked spikes. Their spike times are subject to different transmission delays as well as noise and are hence not simultaneous. A Hebbian learning rule is applied to teach a postsynaptic neuron to exhibit phase-locking by selecting synapses with appropriate delays. The biological background and network setup for this learning process as well as a method for quantifying the quality of learning are introduced.

2.1 Biological background

An incoming sound from one ear is processed in different regions of the brain, which is pictured in figure 1. In the frequency sensitive cochlea the signal is separated into single frequencies. This study focuses on a single frequency channel, the other frequencies being processed analogously. The cochlea passes phase-locked spikes to the nucleus magnocellularis (NM). Spikes in this frequency channel thus only occur around a certain phase of the input signal. However, phase-locking after the cochlea is still imprecise (Maass and Bishop, 2001). After some further processing, the spikes are passed to the nucleus laminaris (NL). Here, the signals from both ears meet. Depending on the azimuthal position of the source of the sound, the signals from the left and right ear arrive in the NL with a certain interaural time difference (ITD). Neurons in the NL are ITD sensitive. Every neuron is positioned such that it receives the signals from both ears coherently for a certain interaural time difference. If the signals arrive at the same time, the neuron's firing rate reaches its maximum (Gerstner et al., 1996). Thus, laminar neurons fulfill the function of coincidence detectors. In further processing steps, the location of the sound source can then be determined depending on which neurons exhibit the highest firing rates. Figure 2 illustrates this schematically. The sound reaches the right ear first and the left ear after the ITD. Both signals are transmitted to the depicted population in the NL. The signal from the right ear has had the same time to travel to the NL as the one from the left ear plus the ITD. Hence, the signals meet at a neuron closer to the left ear, which is marked dark grey in figure 2. This neuron will have the highest firing rate and its position in the NL corresponds to the given ITD. The closer the sound source is to the right ear, the further left the neurons with the highest firing rate will be and vice versa.

However, the ITD is actually a phase difference, because spikes from both ears arrive periodically. In order to locate a sound precisely, very small phase differences need to be distinguished. This means that, firstly, the incoming spikes need to be locked very precisely to their preferred phase and secondly, spikes from different neurons must arrive at the same time. Otherwise, there would be too much noise to resolve small phase differences. The neural network given in Gerstner et al. (1996) can provide such precision. It is located between the NM and the NL, with magnocellular presynaptic neurons stimulating a laminar postsynaptic neuron. The following subsections will explain the properties of this network.

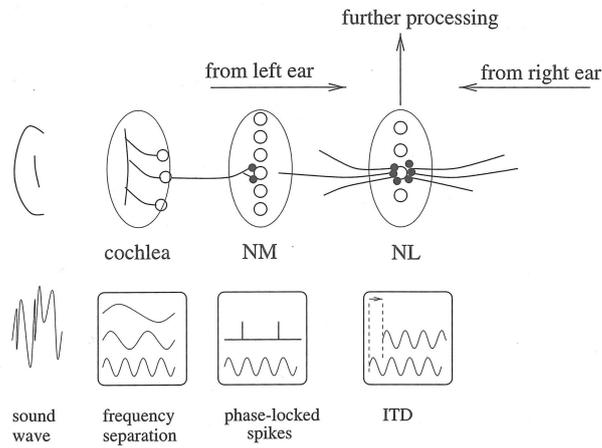


Figure 1: Sound processing from the ear via the nucleus magnocellularis (NM) to the nucleus laminaris (NL). Signals arrive in the NL with a certain interaural time difference (ITD). For details, see text. Taken from Maass and Bishop (2001).

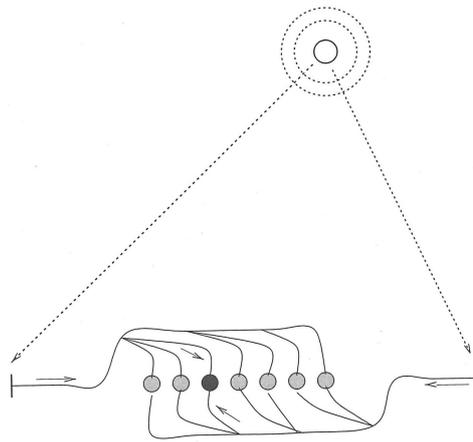


Figure 2: Neurons in the nucleus laminaris which are ITD sensitive. The sounds from both ears are transmitted to the NL and reach the dark grey neuron at the same time. This neuron will have the highest firing rate of the depicted population. Taken from Maass and Bishop (2001).

2.2 Presynaptic input

Presynaptic neurons model a population in the nucleus magnocellularis. It is assumed that some signal processing has already taken place, and that the presynaptic neurons exhibit phase-locked spiking behaviour. To understand how presynaptic spike times are chosen in the network this thesis focuses on, we need to take a closer look at what happens to a single frequency of the input signal when it is transmitted from the cochlea to the presynaptic neurons. After cochlear filtering, an auditory signal with the frequency f is transmitted along the auditory pathway sketched in figure 1 by means of phase-locked spikes. A magnocellular neuron n in the observed frequency channel thus fires periodically around a preferred phase φ_n

of the stimulating signal. This phase will also be referred to as the mean phase of this neuron. However, phase-locking is imprecise at this stage. Spike times of the magnocellular neurons may deviate to a certain extent from their preferred phases. The spike times of a neuron are thus modeled to be Gaussian distributed around its respective φ_n with standard deviation σ_φ . The Gaussian distribution is limited to values within $1\sigma_\varphi$ around φ_n . This jitter accounts for the bandwidth of frequency tuning of the auditory neurons as well as internal sources of noise (Gerstner et al., 1996). The mean firing rate of an auditory neuron is usually much lower than the frequency of its stimulating signal. Therefore, every magnocellular neuron is assigned a spiking probability $p_{spike} < 1$. The value of p_{spike} denotes the probability that a neuron will fire during the current period of the input signal.

However, not all magnocellular neurons of the same frequency channel spike around the same mean phase φ_n . The auditory signal takes different ways as it is transmitted from the cochlea to the neurons in the NM. Consequently, it reaches every magnocellular neuron with a different transmission delay, that is, at a different time. The preferred phase φ_n around which the magnocellular neurons spike to transmit the signal further to the NL thus depends on the transmission delays with which they have received the signal from the cochlea. These transmission delays - and hence the preferred phases φ_n - are assumed to be Gaussian distributed (Gerstner et al., 1996) with a mean μ and standard deviation σ_μ . This distribution will be important throughout this thesis and will be referred to as the *delay distribution* or the *mean phase distribution*. In the discussed network, presynaptic neurons are set to be magnocellular neurons with the properties described above. Every presynaptic neuron has a preferred phase φ_n drawn from a Gaussian distribution and fires periodically around that phase with a maximum jitter of σ_φ and a spiking probability p_{spike} . Figure 3 shows the spike times of an exemplary presynaptic population, with neuron IDs sorted according to their respective mean phase. During every period, a different combination of neurons is active due to the spiking probability which is less than one. The mean phase distribution adapted from Gerstner et al. (1996) is so broad that spikes from up to three different spike volleys overlap. Thus, without tuning of the delays and an according selection of the synapses, phase information could hardly be conveyed.

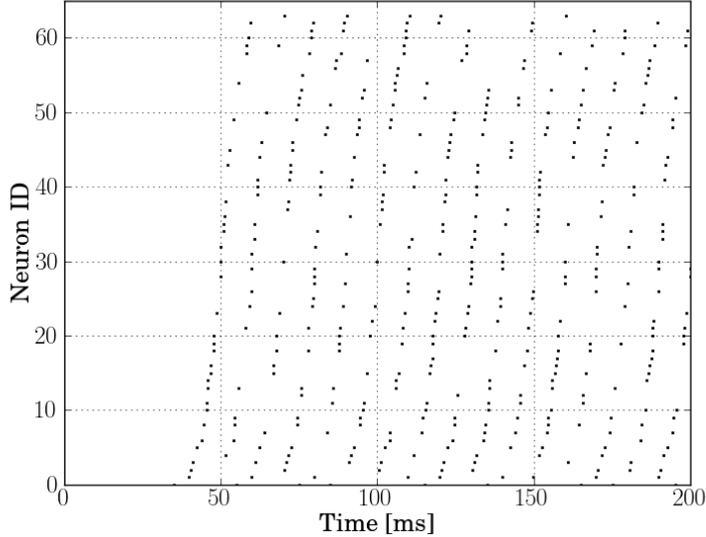


Figure 3: Exemplary spikes of 64 presynaptic neurons. The neuron IDs are sorted according to their respective preferred phase φ_n . These phases are Gaussian distributed around μ with standard deviation σ_μ . Presynaptic spikes volleys occur periodically with frequency f of the auditory signal. Due to the spiking probability $p_{spike} < 1$, different neurons are active during every period.

2.3 Learning process

The presynaptic population is connected to a postsynaptic neuron in the nucleus laminaris. For the postsynaptic neuron, a conductance based leaky integrate-and-fire neuron model is used (Gerstner and Kistler, 2002; Scherzer, 2012). The aim of the learning process is to train the postsynaptic neuron to exhibit phase-locked spikes by selecting synapses with appropriate mean phases φ_n . For this, a Hebbian learning rule is applied. With a time difference $s = t_{post} - t_{pre}$ between the presynaptic (t_{pre}) and postsynaptic (t_{post}) spike, an additive STDP rule is used (Morrison et al., 2008). In Gerstner et al. (1996), it has the form¹

$$W(s) = \begin{cases} [A_+ - A_-] \exp[(s^* - s)/\tau_+] & \text{for } s > s^* \\ A_+ \exp[-(s^* - s)/\tau_+] - A_- \exp[(s^* + s)/\tau_-] & \text{for } s < s^* \end{cases} \quad (1)$$

with A_+ and τ_+ being the amplitude and time constant for causal correlation and A_- and τ_- for anti-causal correlation. The learning window has its maximum at a time difference s^* . Note that s^* is larger than zero in Gerstner et al. (1996). This will be discussed in more detail in section 5.

At the beginning, all pre-post connections have the same synaptic weight w_{start} . Presynaptic input stimulates the postsynaptic neuron and increases the probability of a postsynaptic spike. Let us assume that presynaptic spike volleys cause a postsynaptic spike at a time t_{spike} .

¹In Gerstner et al. (1996), s is defined as $s = t_{pre} - t_{post}$. For consistency, the definition of s as $s = t_{post} - t_{pre}$ is used here since hardware learning windows will be plotted with the same convention. Consequently, the learning window and corresponding equation used here are mirrored around the y-axis compared to Gerstner et al. (1996).

This is illustrated schematically in figure 4. The postsynaptic spike is marked thick black in this figure. Learning then strengthens all synapses which were active shortly before t_{spike} (green area in figure 4) while synapses which were active shortly after t_{spike} are weakened (red area).

Since the spike times of the presynaptic neurons depend on their preferred phase φ_n , the potentiated synapses must all have a similar preferred phase. This can be a phase φ^* shortly before t_{spike} or a phase $\varphi^* \pm nT$ ($n \in \mathbb{N}$), since synapses whose preferred phases differ by a multiple of the period T are active around the same time. Accordingly, spikes in the green area in figure 4 belong to neurons with phases around φ^* or $\varphi^* \pm nT$.

The potentiated synapses have a stronger influence on the postsynaptic neuron. This makes the postsynaptic neuron again likely to fire shortly after these synapses were active. If that is the case, learning will strengthen the already strong synapses most, because their phase corresponds to the phase of the postsynaptic spike (see also Gerstner et al. (1996)). During learning, this leads to synapses parted into a group with maximum synaptic weight and a group with zero weight, the latter group being out of phase compared to the postsynaptic spike times.

As explained above, the phases of potentiated synapses can differ by multiples of T . A histogram of the phases φ_n of the surviving synapses after the learning process will consequently show sharp peaks at a respective distance of T . In Gerstner et al. (1996), synapses are considered to have survived if they have maximum weight. After learning, the postsynaptic neuron only receives input at the phase corresponding that of the surviving synapses. It then fires periodically shortly after that phase. Thus, learning has trained the postsynaptic neuron to exhibit phase-locking.

2.4 Vector Strength

For measuring the quality of phase-locking in the postsynaptic spikes, the *vector strength* is used (Gerstner et al., 1996; Goldberg and Brown, 1969). This metric gives information on how close the postsynaptic spike times are to a common phase. The vector strength can assume values between 0 and 1. A value of 0 would denote evenly distributed spikes with no preferred phase. A value of 1 would mean perfect phase-locking, that is, all spike times are the same modulo a certain period of time. In the case of phase-locking as presented in this study, this is the period T of the stimulating signal with frequency $f = 1/T$. The vector strength is defined as

$$v = \sqrt{X^2 + Y^2} \quad (2)$$

with

$$X = \frac{1}{N} \sum_{i=1}^N \cos(2\pi f t_i) \text{ and } Y = \frac{1}{N} \sum_{i=1}^N \sin(2\pi f t_i), \quad (3)$$

where N is the total number of postsynaptic spikes and t_i is the time of spike i .

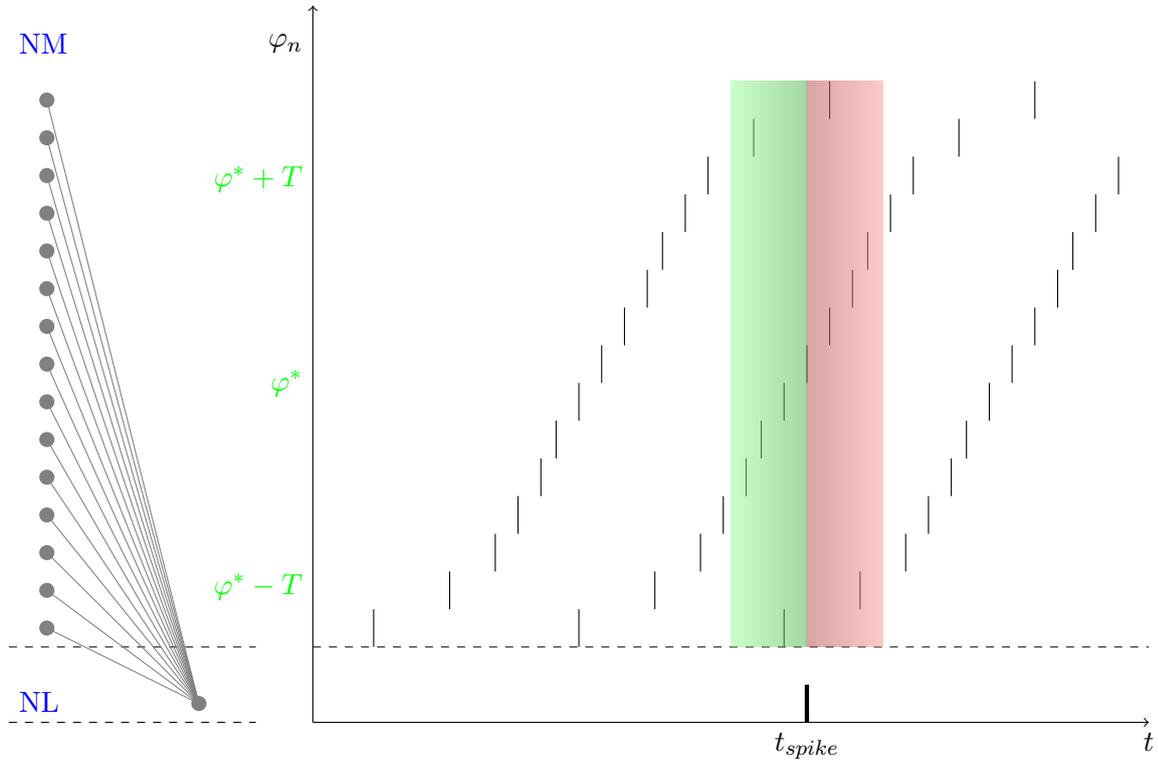


Figure 4: Schematic picture of the network topology and learning process. *Left*: Presynaptic neurons are a population in the NM and connected to a postsynaptic neuron in the NL. *Right*: Presynaptic spikes are plotted, sorted according to their respective preferred phases φ_n . A postsynaptic spike (in the NL) at time t_{spike} is marked thick black. Synapses which were active shortly before the postsynaptic spike are potentiated (green area), synapses which were active after the postsynaptic spike are depressed (red area). If the postsynaptic neuron spiked at a certain phase φ^* , potentiated synapses have a preferred phase shortly before φ^* (middle spike volley) or $\varphi^* \pm nT$ with $n \in \mathbb{N}$ (upper and lower spikes of neighboring volleys).

3 Hardware synapse model

Before the neural network of section 2 is transferred to the Spikey chip, software simulations with a model of the chip’s synapses are run. The model implements some limitations which the hardware synapses have (Pfeil et al., 2012). This is useful for checking whether the network is applicable to Spikey and for predicting the hardware performance in this learning task. The hardware synapses differ considerably from regular STDP synapses (Morrison et al., 2008). The following lists the differences and restrictions of the hardware synapse which are included in the model. In section 5.1, simulations runs are done with this model to analyze the effect these limitations have on learning.

3.1 Synaptic weight resolution

On Spikey, synaptic weights can be set with 4-bit resolution. This limits the weight range that can be used to 16 discrete values. In the hardware synapse model, the maximum hardware weight w_{max} can be set arbitrarily and the weight range $[0, w_{max}]$ is then discretized into 16 values. However, on Spikey, the maximum hardware weight is limited and networks need to be adapted to the available weight range.

3.2 Weight update

A *weight update controller* controls the STDP mechanism and is responsible for reading out synaptic weights. This weight update controller needs 15 ms to process one synapse row on the chip. Since the used network will have 64 synapses with one synapse per row, the total time for one cycle updating all synaptic weights is $t_{proc} = 960$ ms, because the weight update controller processes the synapses sequentially. Thus, the processing time will be almost two orders of magnitude larger than the used STDP time constant. This causes the synaptic weights to be updated with a low frequency and in rather large steps.

Weight updates are determined by the voltages on two capacitors. One is responsible for collecting causal spike pairs, the other for anti-causal spike pairs. The sign of the time difference s between a presynaptic and a postsynaptic spike determines whether the causal or anti-causal capacitor is charged. Its absolute value determines the amount of charge, weighted according to an exponential function where a smaller absolute value of s causes a higher charge. This models the time-dependent aspect of STDP where the weight update for either causality is most pronounced for small absolute values of s (Morrison et al., 2008). The learning window will be explained in more detail in the following subsection.

If the difference between the charge of the causal and anti-causal capacitor is larger than a certain threshold, the synapse is marked for a weight update (Schemmel et al., 2006), which will take place in the next weight update cycle. In that way, a weight update is triggered only by the difference between the two capacitors but not by the total amount of causal and acausal events saved on each capacitor. In the hardware synapse model, the threshold for weight updates can be configured.

3.3 Learning Window

In Gerstner et al. (1996), an STDP learning window of the form in equation 1 with a superposition of exponential terms and different time constants τ_+ and τ_- for causal and anti-causal events is used. The hardware only allows a learning window of the form (see also Schemmel et al. (2006))

$$W(s) = \begin{cases} A_{STDP} \exp[-s/\tau_{STDP}] & \text{for } s \geq 0 \\ -A_{STDP} \exp[s/\tau_{STDP}] & \text{for } s < 0 \end{cases} \quad (4)$$

Therefore, only one time constant τ_{STDP} can be set. This is taken into account for simulations with the hardware synapse model to make the simulations more comparable to the hardware results. The STDP time constants for causal and anti-causal events are therefore set to the same value.

Different STDP mechanisms are realized on hardware by using look-up tables for both the causal and anti-causal events. These determine the weight dependent part of STDP (as opposed to the time-dependent part). For an additive weight update mechanism as it is used here, the look-up table is set such that a weight is updated by a constant amount - to the next higher discrete weight for causal weight update, to the next lower discrete weight for anti-causal weight update. Weight update thus follows the equation

$$w_{new} = w_{old}(1 + W(s)), \quad (5)$$

which was adapted from Schemmel et al. (2006). A comparison of the learning window on the hardware with the one in the original publication can be seen in figure 5. Most noticeably, the learning window in Gerstner et al. (1996) allows a potentiation of synapses which were active very shortly *after* the postsynaptic spike. The yellow area in figure 5 marks those negative time differences s for which the respective synapses will still be potentiated. This is not possible on hardware and will be a cause for discussion in section 5.3.2.

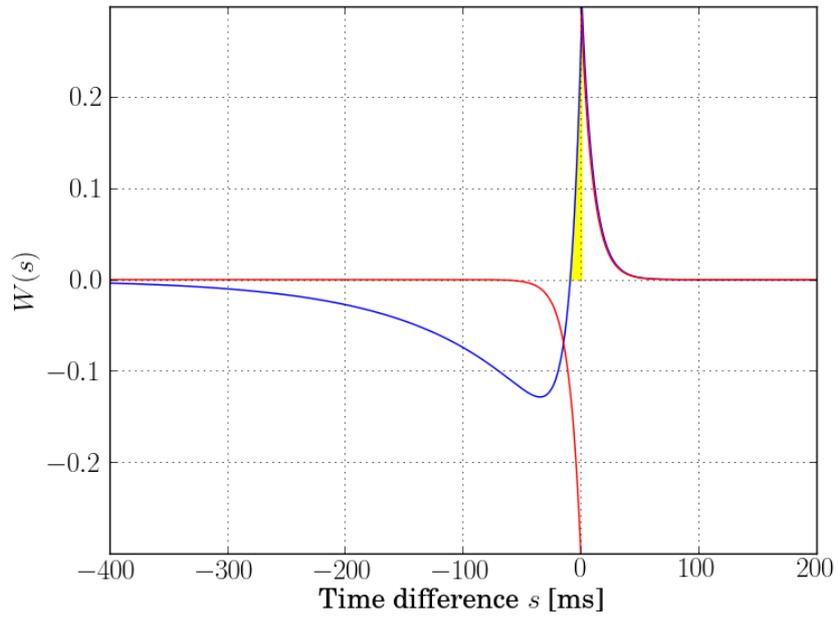


Figure 5: Comparison of the learning window in Gerstner et al. (1996) (blue) and the learning window used for simulations with the hardware synapse model according to equation 4 (red). The original learning window also allows potentiation of synapses which were active shortly after the postsynaptic spikes (yellow).

4 Measurement and adaptation of hardware parameters

The quality of phase-locking in the postsynaptic spiking behavior depends heavily on the neuron, synapse and STDP parameters. These parameters can not be set arbitrarily on the hardware. Consequently, many of the original model parameters in Gerstner et al. (1996) can not be used and have to be adapted to the hardware model and parameter ranges. At the same time, it is necessary to keep the hardware parameters as close as possible to those of the original publication (Gerstner et al., 1996) to allow the comparison of the network performance. This section will focus on different parameters and methods to determine and set all network parameters to an appropriate value.

4.1 Presynaptic stimulus

In Gerstner et al. (1996), a network with 600 presynaptic neurons is used. However, the maximum available number of neurons on the Spikey chip is 192. Since the processing time for one weight update cycle increases linearly with the number of presynaptic neurons (see 3.2), an even smaller network is chosen in favor of a faster weight update. With fewer synapses than in the original network, the postsynaptic neuron receives less input with the consequence of less phase information and learning will take longer. To compensate for this, the spiking probability p_{spike} of the presynaptic neurons is increased.

4.2 Membrane time constant

The hardware membrane time constant τ_m is controlled by a configurable hardware current I_{leak} . A neuron's leakage conductance g_{leak} is proportional to I_{leak} . With

$$\tau_m = \frac{C_m}{g_{leak}} \quad (6)$$

where C_m is the membrane capacity, τ_m is proportional to $1/g_{leak}$ and therefore also to $1/I_{leak}$. The membrane capacity C_m can not be varied on the hardware.

To achieve a certain value of τ_m , a method is necessary to determine the transformation between τ_m and I_{leak} . Applying a synaptic stimulus to the neuron and determining its membrane time constant from the excitatory postsynaptic potential on the membrane would be too imprecise, because synaptic conductances are unknown (Brüderle, 2009). Instead, a method developed by Brüderle (2009) has been used to determine τ_m . The resting potential V_{rest} of the neuron is set *above* its firing threshold V_{thresh} . Consequently, the membrane potential will develop towards V_{rest} and is pulled to the reset potential V_{reset} when it reaches V_{thresh} . The membrane potential then starts developing towards V_{rest} again, following an exponential development. Thus, the neuron fires regularly and its spiking frequency f only depends on τ_m and its refractory period τ_{refrac} , but not on the synaptic conductance because there is no synaptic stimulus.

If V_{thresh} is chosen such that

$$V_{thresh} = V_{rest} - \frac{1}{e}(V_{rest} - V_{reset}), \quad (7)$$

the time interval $1/f$ between two spikes equals $\tau_m + \tau_{refrac}$. It follows that τ_m is

$$\tau_m = \frac{1}{f} - \tau_{refrac}. \quad (8)$$

Thus, τ_m can be determined by simply measuring the spiking frequency of a neuron, with V_{thresh} chosen according to equation 7. However, this method requires knowledge of the respective values of $V_{rest_{HW}}$, $V_{reset_{HW}}$, $V_{thresh_{HW}}$ and τ_{refrac} on the hardware since they may differ from the set ones. They can also be different for every neuron.

In Brüderle (2009), a calibration has been developed using the described method for determining τ_m . It allows to calibrate the membrane time constants of all used neurons to one target value. However, the individual dependency of a neuron's τ_m on I_{leak} has not been examined. This is done here based on the work in Brüderle (2009) and requires the following steps.

Initially, the values V_{rest} and V_{reset} are set. As a first step, the real values of these voltages, $V_{rest_{HW}}$ and $V_{reset_{HW}}$ need to be measured. For V_{rest} , this is done by setting V_{thresh} above V_{rest} to avoid spiking. The measured membrane potential then equals $V_{rest_{HW}}$.

To determine $V_{reset_{HW}}$, the neuron is forced to spike as mentioned above, by setting V_{thresh} to a value $V_{reset} < V_{thresh} < V_{rest}$. The minimum of the resulting membrane potential trace is $V_{reset_{HW}}$, its maximum is $V_{thresh_{HW}}$. The measurement of these values is averaged over several runs.

Analogous to equation 7, the value which $V_{thresh_{HW}}$ should have is calculated to be

$$V_{thresh_{HW}} = V_{rest_{HW}} - \frac{1}{e}(V_{rest_{HW}} - V_{reset_{HW}}) \quad (9)$$

A binary search (Brüderle, 2009) is then started for the value V_{thresh} that has to be set in order to achieve the target $V_{thresh_{HW}}$ that follows from equation 9. When this value has been found, τ_m can be determined as mentioned above, by measuring the spiking frequency of the analyzed neuron. There has not been enough time to integrate a reliable method to measure the refractory period τ_{refrac} . At present, τ_{refrac} has to be determined by reading out the oscilloscope manually.

These steps are then repeated for different values of I_{leak} , recording τ_m every time. With this data, the minimum and maximum possible membrane time constant of every neuron, which is limited by the valid I_{leak} range, is determined. Furthermore, the recorded dependency of τ_m on I_{leak} can be used to set τ_m to a certain target value, which will be done in section 5.2.1.

4.3 Synaptic time constant

Another parameter that needs to be adapted is the synaptic time constant. The shape of an excitatory postsynaptic potential (EPSP) is determined by the membrane time constant τ_m and the synaptic time constant τ_s . The area below an EPSP determines the strength of a synapse and consequently its influence on the postsynaptic neuron. In this case, the time constants need to be chosen such that the EPSP area is not too large. Otherwise, the

EPSPs evoked by spikes of two subsequent presynaptic spike volleys would overlap. If they overlap too much, efficient learning may be impeded. On the other hand, if the synapses are too weak, no or very few postsynaptic spikes will be triggered. This may also result in worse learning and longer burn-in times because less phase information is conveyed to the postsynaptic neuron.

Like the membrane time constant, the synaptic time constant can not be set directly on the hardware. Instead, the shape of an EPSP is controlled by several currents. A current I_{rise} controls the rising ramp of an EPSP while currents I_{out} and I_{fall} control its amplitude and decay time constant. In section 5.2.2, the influence of different values for I_{out} and I_{fall} on the EPSP width and area is examined. Since I_{rise} can have a lot of influence on the voltage ramp amplitude and on the time in which threshold crossing is detected (Brüderle, 2009), this parameter is left at its default value of $1.0\ \mu\text{A}$. The resulting EPSP width and area for different parameters is estimated with a simple setup. A presynaptic stimulus is connected to a postsynaptic neuron. The synaptic weight is chosen such that no postsynaptic spikes are triggered. The behavior of the membrane potential when a presynaptic spike arrives gives information on the EPSP. A comparison of the membrane trace for different values of I_{leak} and I_{fall} and for different synapses can be found in section 5.2.2.

4.4 Learning Window

The plasticity of a synapse is determined by its learning window. In the original publication (Gerstner et al., 1996), an asymmetric learning window with different STDP time constants τ_+ and τ_- for causal and acausal events was used. As mentioned in section 3.3, this is not possible on the hardware. Only one time constant τ_{STDP} can be configured within a certain value range. There are different voltages controlling the shape of the learning window and consequently τ_{STDP} . A voltage V_m controls the temporal stretching of the STDP curve. The capacitors collecting causal and acausal spike events are controlled by voltages V_{clrc} (causal) and V_{clra} (anti-causal). The weight update threshold is determined by two voltages V_{cthigh} and V_{ctlow} , the threshold being proportional to the difference between V_{cthigh} and V_{ctlow} .

The STDP curve can be recorded for every hardware synapse. For this, a presynaptic neuron is connected to a postsynaptic one and pre-post spike pairs are generated with a fixed time difference s . Since postsynaptic spikes can not be set explicitly, they are triggered by another presynaptic trigger population (Pfeil et al., 2012). The time interval between subsequent pre-post pairs is chosen very large to make sure only one capacitor is charged, i.e. no causality is detected between the last spike of one pair and the first spike of the next pair. For different values of s , the number of pre-post spike pairs N that is necessary to trigger weight update is measured. The STDP curve can then be plotted as $1/N$ versus the time difference s . In section 5.2.3, STDP curves are recorded for every synapse allocated for the used network. Also, the influence of the different voltages on the learning window is examined and an appropriate learning window for further hardware emulations is chosen.

4.5 Weight recording

On hardware, recording synaptic weights during learning is not implemented. They can only be read out before and after a simulation. Since the weight development of different synapses with respect to their preferred phases is interesting for the learning process, the following method has been used to estimate it. A learning process with the duration t_{run} is split into several runs with increasing durations from 0 to t_{run} and the synaptic weights after learning are recorded for every t_{run} . The same mean phase distribution is used for every emulation, but the presynaptic spike times still differ slightly because the jitter σ_φ around the mean phase φ of a neuron is generated anew every time. Since the learning behavior of the postsynaptic neuron with regard to its preferred phase and firing rate differs from run to run, several emulations are done for every t_{run} and the recorded weights are averaged. Finally, the mean weight is plotted for every synapse versus the respective learning time. This method is not as accurate as direct weight recording during the learning process, but gives a good idea of how synaptic weights develop with regard to the corresponding mean phases and will be used in section 5.3.2.

5 Results

5.1 Software Simulations

The hardware synapse model described in section 3 is used for the network to test the quality of phase-locking with all restrictions this synapse offers. The parameters used for simulations with this model and the parameters used in the original publication Gerstner et al. (1996) are listed in table 1. Compared to Gerstner et al. (1996), the learning process runs on a time scale stretched by a time factor x . This is done to make sure all time constants are within their respective hardware value range. Since hardware emulations will be run on the same time scale, this makes simulations in this section more comparable to the hardware results in section 5.3. The choice of the used factor $x = 20$ is motivated in section 5.2.1. All times mentioned here and in the following sections are given in biological real-time for a better comparison to the results in Gerstner et al. (1996).

Time factor x	20	1
Number of synapses N	64	600
Input frequency f [Hz]	100	2000
Simulation duration t_{run} [s]	200	3000
<i>Presynaptic population</i>		
Mean delay μ [ms]	50.0	2.5
Standard deviation of delays σ_{μ} [ms]	6.0	0.3
Jitter around preferred phase σ_{φ} [ms]	0.8	0.04
Spiking probability P_{spike}	0.5	0.14
<i>Postsynaptic neuron</i>		
Membrane time constant τ_m [ms]	2.0	0.1
Resting potential V_{rest} [mV] ²	-65.0	-
Reset potential V_{reset} [mV] ²	-80.0	-
Threshold potential V_{thresh} [mV] ²	-45.0	-
Refractory period τ_{refrac} [mV] ²	0.0	-
<i>Synapse properties</i>		
Synaptic time constant τ_s [ms]	2.0	0.1
STDP time constant τ_{STDP} [ms]	10.0	0.5
Initial weights w_{start} [μ S] ³	0.15	1
Maximum allowed weight w_{max} [μ S] ³	0.24	3
Processing time for one synapse row [ms]	15	-
Synapses per line	1	-

Table 1: Parameters for simulations with the hardware synapse model compared to the original parameters. Parameters used in this thesis are listed in the left column, the reference parameters from Gerstner et al. (1996) in the right column.

²The parameters V_{rest} , V_{reset} , V_{thresh} and τ_{refrac} were not given in the original publication (Gerstner et al., 1996).

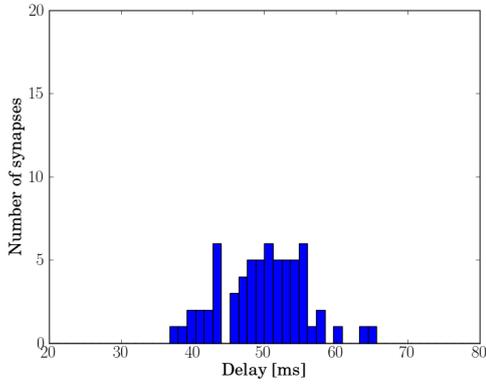
³For synaptic weights, no units were given in the original publication.

The number of synapses N has been lowered for faster weight updates. To compensate for the decreased input, the spiking probability of the presynaptic neurons has been increased (see section 4.1). In Gerstner et al. (1996), approximately 80 of the 600 synapses are active during each period of the input signal, implying $P_{spike} \approx 0.14$, whereas here it has been chosen to be $P_{spike} = 0.5$, resulting in about 32 active synapses per period. Due to the changed presynaptic input, the initial weights w_{start} also had to be adapted to provide a sufficient stimulus for the postsynaptic neuron (Scherzer, 2012). Figures 6 and 7 show the results of an exemplary learning process with the parameters given in table 1.

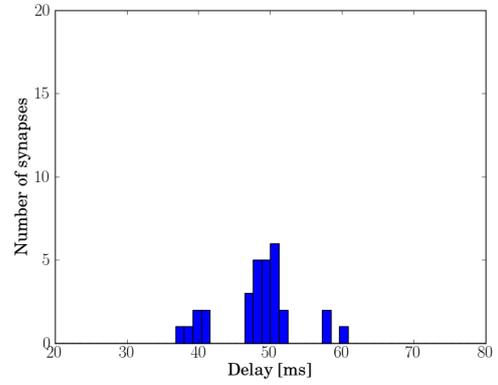
Figure 6(a) shows the delay distribution of the 64 synapses before learning. As explained in section 2.2, this distribution equals the distribution of the preferred spiking phases φ_n of the presynaptic neurons. The delays of the 30 synapses which have survived after 200 s of learning are shown in figure 6(b). A synapse is considered to have survived if its weight exceeds the initial weight w_{start} . As expected (see section 2.3), learning selected those synapses whose delays differ by multiples of the period T of the input signal. The delay histogram thus shows sharp peaks at a respective distance of $T = 10$ ms, meaning that survived synapses have preferred phases differing by multiples of T . Figure 6(c) shows a histogram of the postsynaptic spikes modulo T . All spikes throughout the whole runtime have been taken into account for this plot. A sharp peak in the figure clearly shows the phase-locked spiking behavior of the postsynaptic neuron. The vector strength corresponding to this plot is $v = 0.83$, the mean firing rate of the postsynaptic neuron is 62 Hz.

In figure 6(d), the weight distribution of all synapses after learning is illustrated. The x-axis covers the possible weight range on hardware. The green line marks the mean initial weight w_{start} , whereas the red line marks the mean weight calculated from the weight distribution after learning. As discussed in section 2.3, learning has led to a bimodal weight distribution with a group of synapses with zero weight and a group with the maximum weight w_{max} . The weight development of the 64 synapses during learning is shown in figure 7. It can be seen that weights are updated stepwise and in fairly large time intervals. The plot confirms that synapses which are already strong are potentiated even further because they have more influence on the postsynaptic neuron’s spiking behavior and are thus in phase with its spikes (section 2.3). Weak synapses with a mean phase differing too far from the phase of the postsynaptic spikes are in turn further weakened.

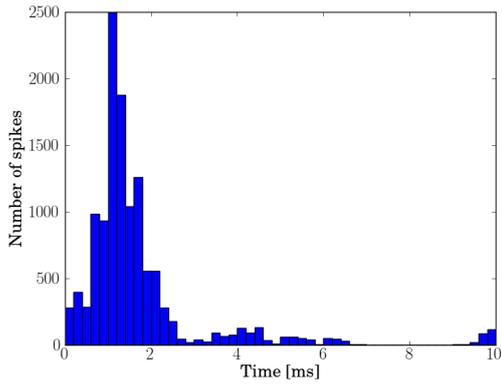
For comparison, the accumulated results of 20 independent runs are depicted in figure 8. The simulation parameters are taken from table 1, but the mean phase distribution and hence the presynaptic spike times are generated anew for every run. Figure 8(a) shows the accumulated delays, i.e., the mean phase distribution, before learning. The Gaussian shape is now more pronounced than for a single run. Figure 8(b) shows the delays of all surviving synapses, again with peaks at a respective distance of $T = 10$ ms. A mean number of 35 synapses have survived learning. A histogram of the postsynaptic spike times modulo T in figure 8(c) shows a very broad peak compared to a single measurement (figure 6(c)). This is due to the fact that the mean phase distribution of the presynaptic neurons is different for every run. Consequently, the preferred spiking phase of the postsynaptic neuron may vary for different runs. The peak in figure 6(c) is accordingly broader than the corresponding result of a single run. However, the mean vector strength of the 20 runs used for this plot is $\bar{v} = 0.87$, confirming good phase-locking in the single runs. Note that \bar{v} was not calculated from the accumulated spikes in figure 8(c). It is the mean of the resultant values of v after



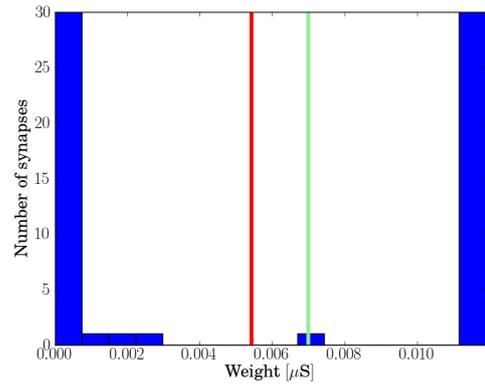
(a) Delay distribution before learning



(b) Delays after learning



(c) Postsynaptic spike times modulo T



(d) Synaptic weights after learning

Figure 6: Exemplary result of a 200 s learning process. Figure (a) shows the delay distribution of the 64 synapses before learning, figure (b) the delays of the 30 surviving synapses. The postsynaptic spikes times are plotted modulo T in figure (c). All spikes throughout the learning process have been taken into account. The mean firing rate of the postsynaptic neuron is 62 Hz, the vector strength is $v = 0.83$. Figure (d) shows the weight distribution after learning. The green line marks the initial weights w_{start} , the red line marks the mean weight after learning.

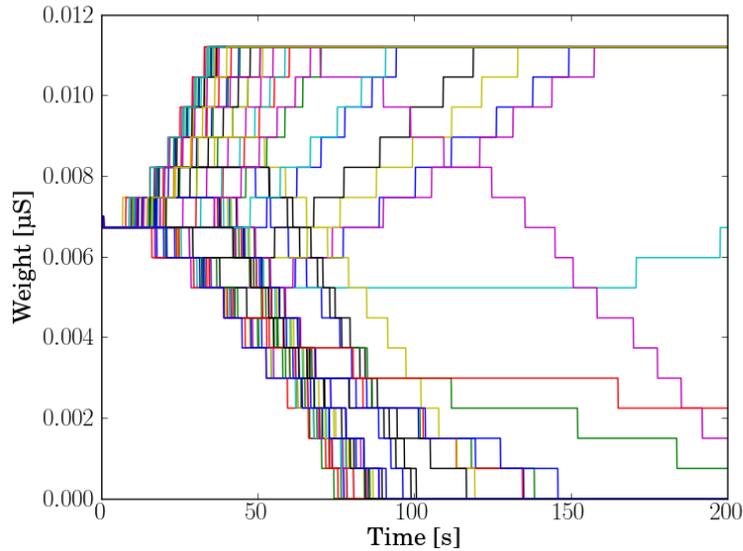
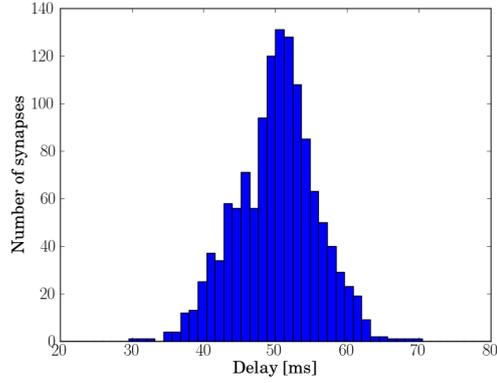


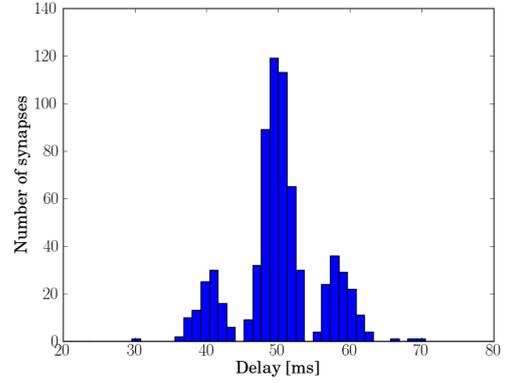
Figure 7: Weight development of the 64 synapses in the network during 200 s of learning. After learning, the synapses are mostly divided into two groups with zero or maximum weight, respectively.

every single run. The accumulated weight distribution after learning is depicted in figure 8(d). Again, there is a distinctly bimodal weight distribution with only very few synapses not having minimum or maximum weight. The mean weight after learning is comparable to a single run.

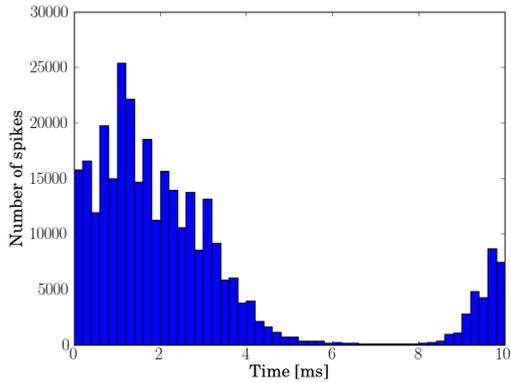
These results lead to the conclusion that, if the model parameters are adapted to the hardware capabilities, phase-locking works very well. Even though the hardware synapse offers many restrictions, the resulting delay and weight distribution meet the expectations and precise temporal spiking behavior of the postsynaptic neuron can be achieved. This suggests that the network is applicable to Spikey if all parameters are set such that they are as close as possible to the ones used here. The following sections will therefore focus on the measurement and appropriate setting of these parameters.



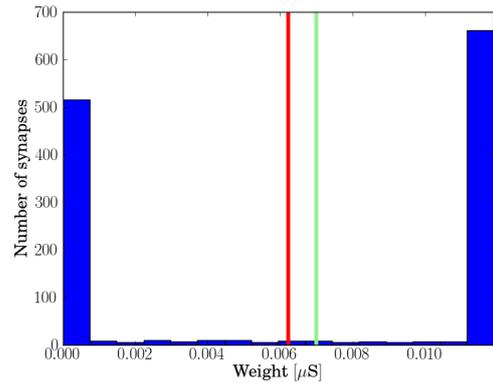
(a) Delays before learning



(b) Delays after learning



(c) Postsynaptic spike times modulo T



(d) Synaptic weights after learning

Figure 8: Accumulated results of 20 independent simulation runs. Figure (a) shows the Gaussian distribution of all delays before learning, figure (b) the surviving synapses after 200 s of learning. The accumulated postsynaptic spike times modulo T are shown in figure (c) whereas the weight distribution after learning is illustrated in figure 8(d). The green line marks w_{start} , the red line the mean weight of all runs after learning.

5.2 Hardware parameter modification

5.2.1 Membrane time constant

As a first step towards transferring the network onto Spikey, the membrane time constant τ_m of the postsynaptic neuron needs to be adjusted. The dependency of τ_m on the current I_{leak} is recorded as described in section 4.2. The parameters V_{rest} and V_{reset} are set to the values listed in table 1, and I_{leak} is varied from $0.01 \mu\text{A}$ to the maximum value $2.5 \mu\text{A}$. The refractory period τ_{refrac} is assumed to be 0 ms since it was set to the minimum possible value and measurements with the oscilloscope show no distinct refractory period for the set parameters.

Measurements of the individual $V_{rest_{HW}}$ and $V_{reset_{HW}}$ are averaged over 3 runs for each I_{leak} value, $V_{thresh_{HW}}$ is averaged over 5 runs. Since the determination V_{rest} , V_{reset} and V_{thresh} requires recording the membrane trace with the oscilloscope, more runs would cost too much time and have not been done here. The membrane time constant is read out once with the measured parameters. In figure 9, the measured τ_m for 50 different values of $1/I_{leak}$ is plotted for 10 neurons on Spikey chip 445. Since τ_m should be proportional to $1/I_{leak}$ according to equation 6, a linear dependency would be expected. However, the slope of the recorded curves decreases with $1/I_{leak}$. This might be attributed to the fact that the measured V_{rest} and V_{reset} are different for every neuron, but the specific choice of V_{thresh} in equation 9 makes the measurement of the membrane time constant independent from the individual V_{rest} and V_{thresh} . Thus, the recorded neurons all show a similar, non-linear dependency on the inverse of I_{leak} .

Additionally, the resting potential V_{rest} has been found to show a distinct dependency on I_{leak} . Figure 10 shows the resting potential of the same 10 neurons versus the inverse of I_{leak} . With increasing $1/I_{leak}$, the recorded V_{rest} decreases. Similar curves, which are not shown here, have been measured for all other neurons. The reset potential V_{reset} , apart from some fluctuations, remains the same, independently of I_{leak} .

The exemplary curves in figure 9 suggest that the value range for τ_m is quite limited on Spikey. Again, this varies for different neurons, the neurons with the topmost curves allowing τ_m values from approximately 3 to 11 ms, whereas others only allow values from 3 to about 6 ms. This heterogeneity of the network neurons makes a common dependency of τ_m on I_{leak} hard to find. Instead, the recorded data (see figure 9) for the individual neurons provides the possibility to determine the minimum and maximum membrane time constant of a neuron and to estimate I_{leak} for a target τ_m or vice versa.

For better comparison of hardware emulations of the network with the results in Gerstner et al. (1996), parameters as close as possible to the original network parameters need to be chosen. Since the original membrane time constant is $\tau_m = 0.1 \text{ ms}$, the neuron with the smallest possible membrane time constant is the most suitable one. On Spikey 445, this is the neuron with ID 198, which is marked this black in figure 9. The selection of this postsynaptic neuron also determines the time factor x by which all time constants are stretched compared to Gerstner et al. (1996). With a minimum possible τ_m of about 2 ms and the original value being 0.1 ms , this constitutes $x = 20$ for all further parameter adaptations.

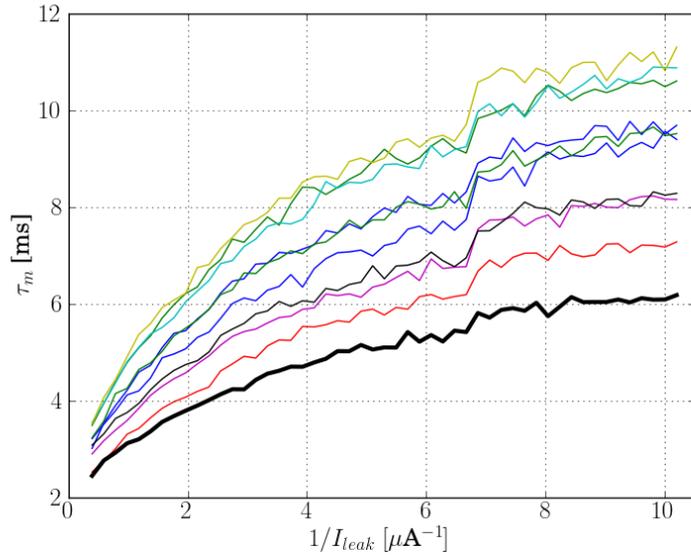


Figure 9: Dependency of τ_m on $1/I_{leak}$ for neuron 192 to 201. Ideally, there should be a linear dependency. The thick black curve marks the results for neuron 198, which is chosen as a postsynaptic neuron for later hardware emulations. Every other color marks the recorded data for a certain neuron.

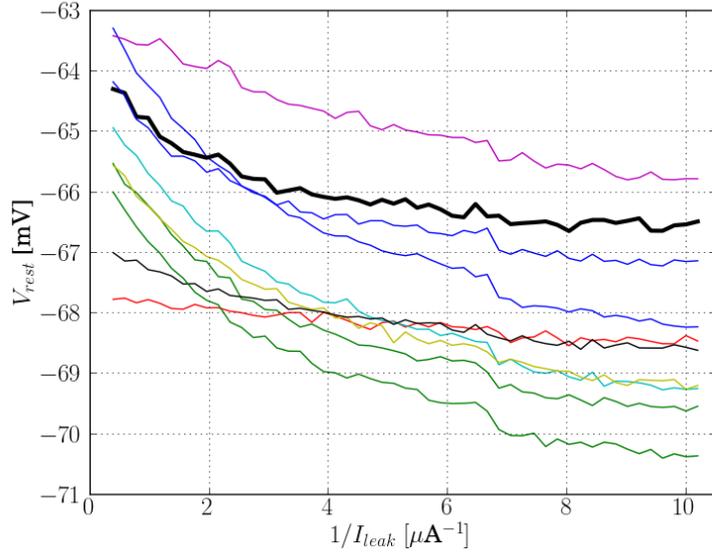


Figure 10: Resting potential V_{rest} versus $1/I_{leak}$ for neuron 192 to 201. Again, the thick black curve marks the measured values for neuron 198.

5.2.2 EPSP modification

In the following, the chosen neuron 198 is stimulated by regular presynaptic input via one synapse. The parameters V_{rest} , V_{reset} and V_{thresh} are set as in table 1. Then, the currents controlling the EPSP shape are set to different values, which are listed in table 2. Figure 11 shows which influence I_{out} and I_{fall} have on the membrane potential of the postsynaptic neuron. This data is not suitable for quantitative estimations, however, because the strength of the synapses varies on the chip and very different EPSPs may be evoked by different synapses. Furthermore, the data has not been recorded during in the postsynaptic neuron's high conductance state where it may integrate differently. Qualitatively, the influence of I_{fall} on the decay time constant is clearly visible in figure 11. The smaller I_{fall} is chosen, the steeper the slope of the curve after its maximum is. However, I_{fall} also influences the amplitude of an EPSP as can be seen by comparing the blue and the green curve, for instance. A comparison of the green and the black curve shows that a higher I_{out} also results in a higher amplitude of the EPSP.

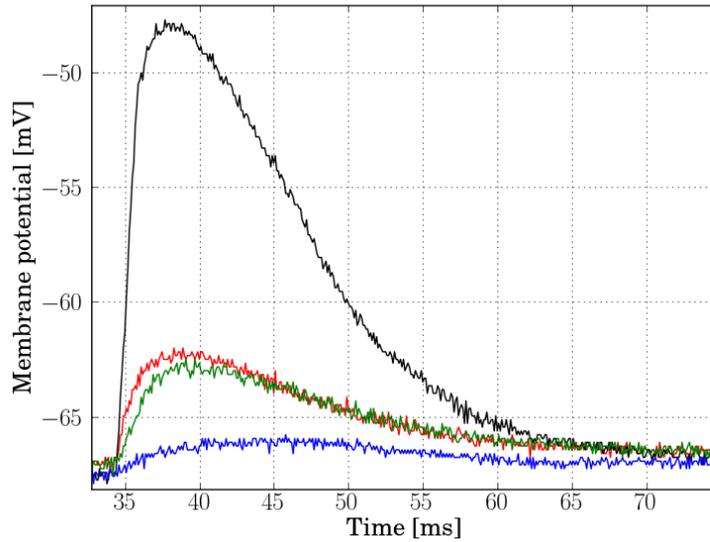


Figure 11: EPSPs for different values of I_{out} and I_{fall} , which are listed in table 2.

Color	I_{out} [μ A]	I_{fall} [μ A]
black	2.5	0.5
red	2.5	2.5
green	0.5	0.5
blue	0.5	2.5

Table 2: Parameters used for the curves in figure 11.

For the learning process, an EPSP with a reasonably small area is needed, because the postsynaptic neuron receives input with high rate. With the time stretching factor $x = 20$ determined in section 5.2.1, a small EPSP with half maximum width of about 5 ms and a synaptic time constant $\tau_s = 2$ ms would be required, given the original parameters in Gerstner

et al. (1996). Since these values could not be determined precisely, $I_{out} = I_{fall} = 2.5 \mu\text{A}$ was chosen for the hardware emulations. In figure 11, the EPSP for these values still has a reasonably small τ_s (as opposed to the blue one) while its area is not too large.

For the chosen parameters, EPSPs for 100 synapses of the used neuron are recorded with the same setup. Figure 12 shows the results of three runs for every synapse. The black curve marks the mean of all synapses, the red area its error. As expected, there is a high variation of the EPSP size and shape for different synapses. This will be taken into account for discussions about the quality of the hardware results in section 6.

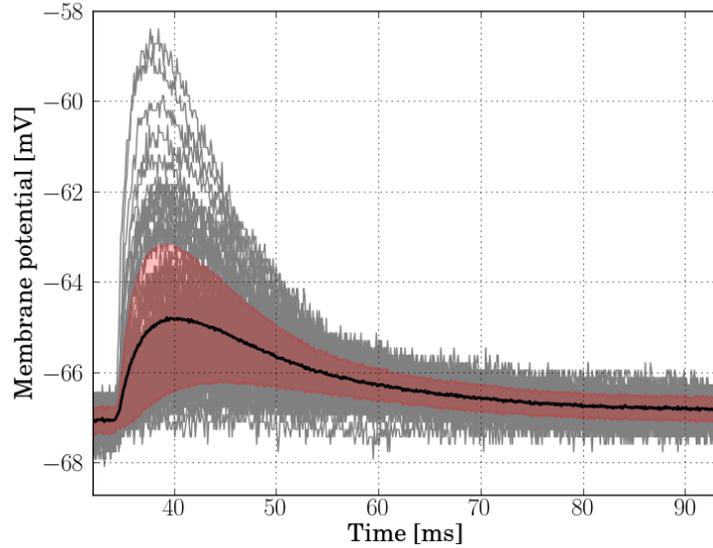


Figure 12: EPSPs for 100 synapses of neuron 198 (grey curves) and their mean (black curve) with error (red area). The values for every synapse have been averaged over 3 runs.

5.2.3 STDP curves

A learning window close to the model (Gerstner et al., 1996) needs to be selected for the learning process. Since the membrane time constant for the chosen neuron is $\tau_m \approx 2$ ms, the STDP time constant should be around $\tau_{STDP} \approx 10$ ms to keep the relations between the time constants as proposed in Gerstner et al. (1996). Learning windows are recorded with different STDP parameters to find suitable values. The results are shown in figure 13 with the corresponding parameter sets listed in table 3. The values in this table are chosen such that the value range within which a reasonably symmetric STDP window can be achieved is covered for every parameter. The thick black curve, hence also called the reference curve, marks the learning window which will be chosen for the learning process for reasons explained below. The other curves are the result of variations of certain parameters compared to the reference parameter set.

The parameter V_m is responsible for the temporal stretching of the STDP window. The red curve in figure 13 has been recorded with $V_m = 0$ V whereas $V_m = 0.3$ V is applied for all other curves. Consequently, the red curve is much smaller in temporal matters than the other curves.

The voltages V_{clrc} and V_{clra} control the capacitors which collect the causal and acausal spike events. Due to circuit design, V_{clra} always needs to be chosen a bit higher than V_{clrc} for a symmetric learning window, if V_m is chosen larger than 0 V. Increased values for V_{clrc} and V_{clra} compared to the reference parameters (see green curve in 13) result in more charge on the respective capacitors. For small time differences s between the presynaptic and postsynaptic spike, this results in a broad plateau region of the STDP curve in which a single pre-post spike pair is sufficient to trigger a weight update ($1/N = 1$). This plateau region accordingly is less pronounced for decreased values of V_{clrc} and V_{clra} . The yellow curve has been recorded with smaller values for V_{clrc} and V_{clra} and the plateau region has vanished, but the slope of the curve has hardly changed. This learning window triggers weight updates only for fairly small values of s , while many pre-post spike pairs are needed to trigger weight updates for larger time differences.

The weight update threshold (see section 3.2) can be set with the voltages V_{cthigh} and V_{ctlow} . A small difference $V_{cthigh} - V_{ctlow}$ means a low threshold. In figure 13, the blue curve marks a lower threshold than the one set for the reference curve. This results curves comparable to the ones for higher V_{clrc} and V_{clra} values, because only one pre-post spike pair is needed to trigger weight updates.

If $V_{cthigh} - V_{ctlow}$ is chosen larger compared to the reference curve, more pre-post spike pairs are needed to trigger weight updates and the plateau region shrinks again (grey curve). However, the slope of the curve largely remains the same. A change of the slope could not be achieved without a loss of symmetry of the STDP curve. This arises some problems regarding τ_{STDP} . Since the slope of the curve can not be varied greatly, τ_{STDP} does not seem adaptable to the model. However, the plateau region appearing for larger $V_{clrc/a}$ values or for a smaller $V_{cthigh} - V_{ctlow}$ makes it difficult to determine how to actually measure τ_{STDP} .

Since time was limited for a detailed investigation of the plateau region, the STDP curve for hardware emulations is chosen for the following criteria: The value $\tau_{STDP} = 10$ ms which corresponds to $\tau_m \approx 2$ ms can not be set. As far as it can be estimated from the plots, τ_{STDP}

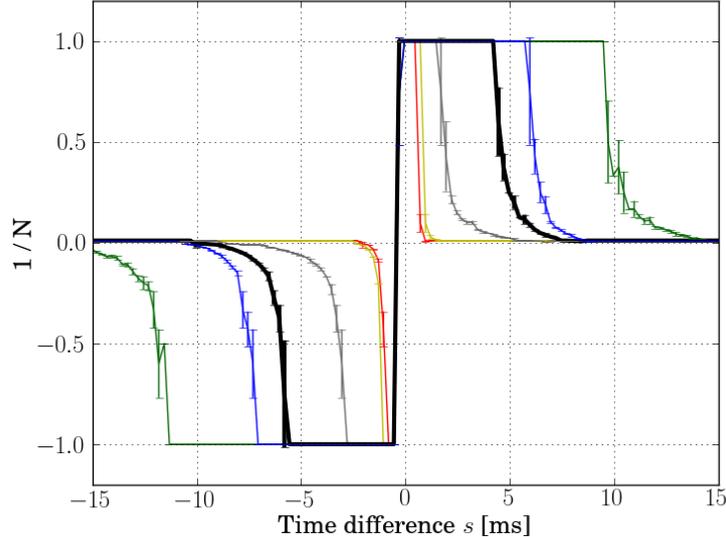


Figure 13: STDP curves of a single synapse with variation of the STDP parameters V_{clrc} , V_{clra} , V_{cthigh} , V_{ctlow} and V_m . The thick black curve points out the parameters used for the learning process. A list of the set values for every curve can be found in table 3.

Color	V_m [V]	V_{clra} [V]	V_{clrc} [V]	V_{cthigh} [V]	V_{ctlow} [V]
thick black (reference curve)	0.3	1.23	1.20	1.00	0.80
blue	0.3	1.23	1.20	0.90	0.82
grey	0.3	1.23	1.20	1.20	0.70
green	0.3	1.30	1.27	1.00	0.80
yellow	0.3	1.03	1.00	1.00	0.80
red	0.0	0.96	0.90	1.00	0.80

Table 3: STDP parameters used for the curves in figure 13.

is too small. A certain plateau region is therefore accepted for the used learning window to enable weight updates for larger absolute values of s , too. On the other hand, the plateau region must not be too broad because all time differences s between 0 and the end of the plateau would then experience the same weight update. Thus, learning would also select synapses whose delays differ too much from the preferred phase of the postsynaptic neuron, resulting in wider peaks in the delay histogram after learning and less phase selection. STDP curves for different synapses may also vary due to fixed-pattern noise. Therefore the values for V_{clrc} , V_{clra} , V_{cthigh} and V_{ctlow} are chosen such that they provide a trade-off between a large enough τ_{STDP} and a plateau region which is not too broad. The thick black curve in figure 13 and its parameters listed in table 3 are used for the network.

With this set of parameters, STDP curves of 100 of the synapses of neuron 198 are recorded. Figure 14 shows grey curves for the different synapses. As expected, the STDP curves vary. Some curves are very asymmetrical or differ a lot from the mean curve. These synapses are blacklisted, meaning that they are not used for the network. With a total of only 64

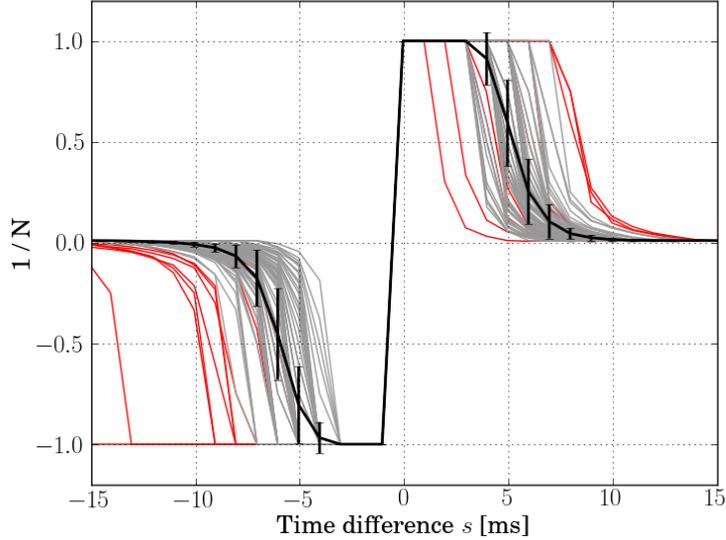


Figure 14: STDP curves of 100 synapses (grey) and their mean value with error (black). Synapses which are blacklisted in the learning process because their learning windows differ too much from the mean curve are marked red.

synapses, they might have a distorting influence on the learning behaviour. As weight update is processed row-wise, blacklisted synapses are processed too, even if they do not have any influence in the network, resulting in a higher weight update frequency. This must be taken into account when choosing how many synapses to blacklist. In figure 14, blacklisted synapses are marked red. Note that the causal/anti-causal branch of a curve can be marked red even though it is close to the mean because its respective anti-causal/causal branch is too far off. A total of 20 synapses out of the recorded 100 is blacklisted, which is 20 %. The black curve in figure 14 shows the mean learning window of the synapses which were not blacklisted. Out of these synapses, 64 are used for hardware emulations of the given network. Between these 64 used synapses, there are 18 blacklisted ones. This increases the processing time for weight update to

$$t_{proc}^* = (64 + 18) \cdot 15 \text{ ms} = 1230 \text{ ms} \quad (10)$$

or a weight update frequency of $f_{proc} = 1/t_{proc}^* = 0.81 \text{ Hz}$, which is very slow compared to the input frequency of 100 Hz. The results of hardware emulations of the used network are given in the following sections.

5.3 Phase-locking on hardware

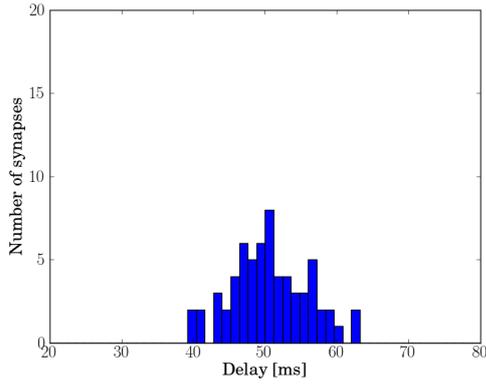
After the adaptation of the membrane time constant, the EPSP width and the learning window to the model proposed by Gerstner et al. (1996), the network is run on Spikey. The network topology and runtime are the same as in simulations with the hardware synapse model in section 5.1. The experimental results and an analysis of phase-locking on hardware are given in the following sections.

5.3.1 Learning results

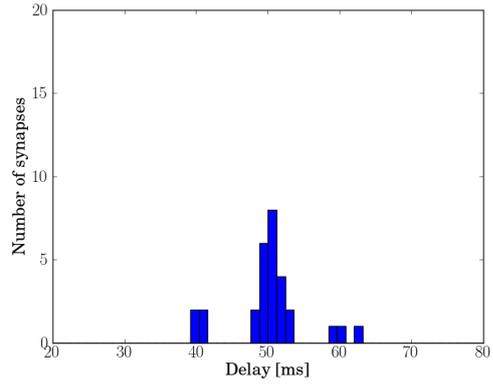
As in section 5.1, a network with 64 presynaptic neurons all connected to a postsynaptic neuron is used. The parameters remain the same as in table 1, with the exception of the synapse properties which have been adapted in the preceding sections. The initial synaptic weight w_{start} is set to the hardware value 2 of 16. Higher initial weights have been found to result in too high mean firing rates of the postsynaptic neuron with the consequence of very poor learning.

The results of a single exemplary run can be seen in figures 15 and 16. The plots are analogous to figure 6, with the delays of the synapses before learning in 15(a), the delays of the surviving synapses in figure 15(b) and the postsynaptic spikes modulo T and the weight distribution after learning in figures 15(c) and 15(d), respectively. The runtime in biological real-time has been 200 s as in the software simulations in section 5.1. 29 synapses have survived learning. Again, synapses are considered to have survived if their weight after learning exceeds w_{start} . As in the software simulations, the delay distribution of the surviving synapses shows sharp peaks. Synapses whose delays differ by multiples of $T = 10$ ms have been selected by the learning rule. The phases of the postsynaptic spikes, which are depicted in figure 15(c) show a pronounced peak. The postsynaptic neuron has been trained to exhibit phase-locking with a mean firing rate of 89 Hz and a vector strength of $v = 0.91$. Unlike the software simulations, however, the weight distribution after learning, which is shown in figure 15(d), is not distinctly bimodal. A certain splitting up of the synaptic weights has taken place, but not all synapses have a weight that is either minimal or maximal. The development of this weight distribution will be discussed in more detail in section 5.3.2.

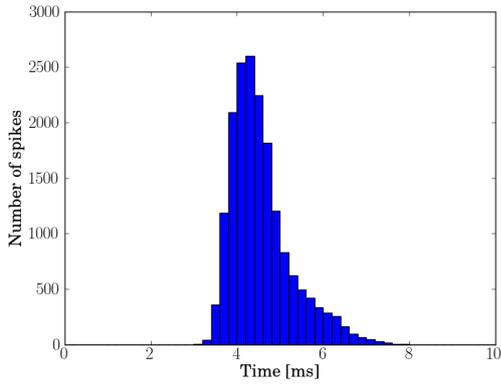
Figure 16 gives an impression of how the learning process evolves. It shows the development of the postsynaptic neuron’s mean firing rate during the learning process. At first, due to the low initial weight w_{start} of all synapses, there are very few postsynaptic spikes. Those synapses which were active before the postsynaptic spikes are then strengthened and consequently have a larger impact on the postsynaptic neuron, increasing its firing rate further. This results in a faster and faster increase of the mean firing rate in the first 20 s (figure 16(a)). Due to the influence of the strongest synapses, a preferred spiking phase of the postsynaptic neuron has already manifested itself. This leads to depression of the synapses which fire shortly after the phase of the postsynaptic spikes. Thus, the total presynaptic input is not increased as rapidly anymore and the mean firing rate slowly develops towards a constant value of 89 Hz after about 150 s (figure 16(b)). This *burn-in time* is in good accordance with the results in section 5.1. In the software simulations, most of the synaptic weights have developed towards their final values by that time, too (see figure 7).



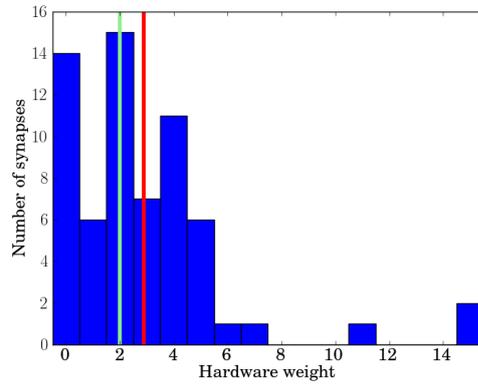
(a) Delays before learning



(b) Delays of the surviving synapses after learning

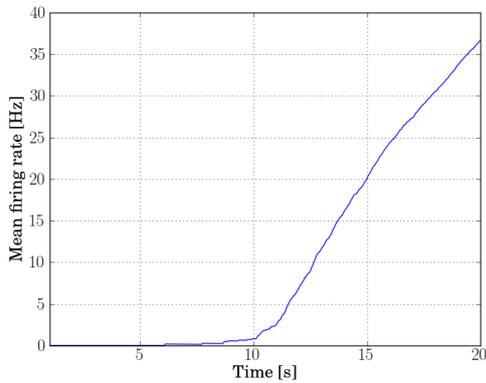


(c) Postsynaptic spike times modulo T

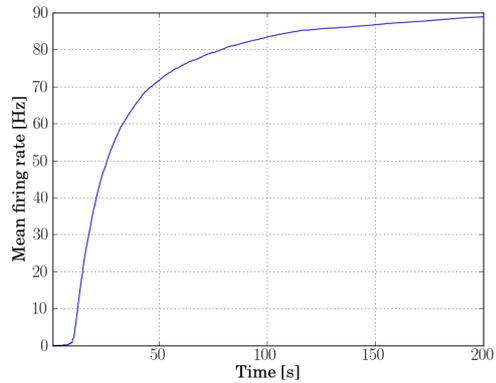


(d) Synaptic weights after learning

Figure 15: Exemplary result of a 200s hardware emulation of the network from Gerstner et al. (1996). Figure (a) shows the delay distribution of the 64 synapses before learning, figure (b) the delays of the 29 synapses which survive learning. The postsynaptic spike times modulo T are depicted in figure (c). The mean firing rate is 89 Hz, the vector strength is $v = 0.91$. Figure (d) shows the synaptic weight distribution after learning.



(a) First 20s



(b) Total run time

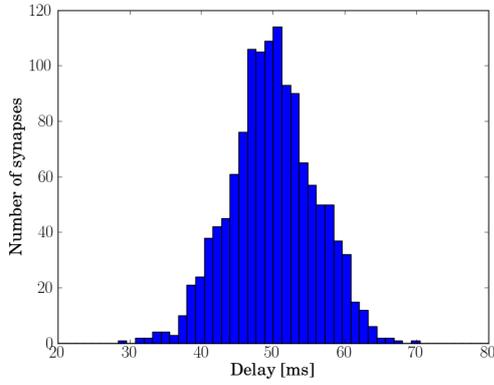
Figure 16: Development of the postsynaptic neuron's mean firing rate during 200 s of learning.

For better statistics, the accumulated results of 20 independent runs are shown in figure 17. For every run, a new distribution of the preferred phases φ_n and new presynaptic spike times have been generated. Figure 17(a) shows the Gaussian delay distribution of all synapses before learning. After 200 s of learning, a mean number of 29 synapses have survived. Their accumulated delays after learning are plotted in figure 17(b). Peaks spaced at intervals of $T = 10$ ms occur here, too. The spike times of the postsynaptic neuron modulo T are plotted in figure 17(c). As in the accumulated software simulations (see figure 8), the resultant peak is broader than the one of a single measurement due to different spiking phases of the postsynaptic neuron in each run. The mean vector strength is $\bar{v} = 0.87$ and thus equals the mean vector strength for the software simulations in section 5.1. Again, \bar{v} has been calculated from the measured vector strength after every single run. The mean firing rate for all runs is 95 Hz. This value differs significantly from the mean firing rate of 62 Hz measured in software simulations. The reason for this is the limited maximum hardware weight w_{max} . As mentioned above, a low value of 2 is chosen for the initial weight w_{start} in order to avoid too high firing rates. However, during learning, some synapses are strengthened, leading to a higher mean weight than w_{start} after learning. The green and red lines in figures 15(d) and 17(d), marking the mean weight before and after learning, show this. Consequently, the postsynaptic neuron receives a higher total presynaptic input for this weight distribution than at the beginning and hence exhibits a higher firing rate. This is different to the software simulations in section 5.1, where the mean weight develops towards lower weights than w_{start} during learning. To achieve a weaker total presynaptic input and accordingly lower firing rates on hardware, a calibration towards lower weights would be necessary. This will be discussed in more detail in section 6.

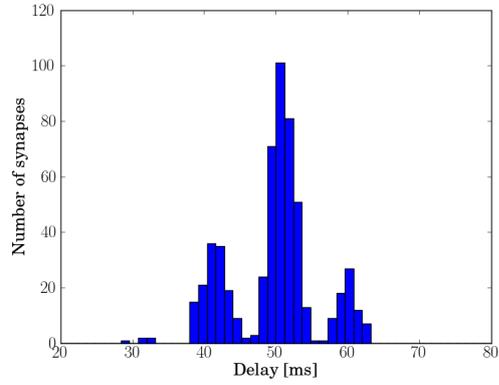
5.3.2 Weight development

The weight development on hardware during learning is shown in figure 18. Each curve marks the weight of one synapse after a certain learning time t_{run} , each with a different mean phase φ_n . The curves have been recorded as described in section 4.5, with the same φ_n distribution, but with different spike jitters σ_φ . Synaptic weights have been recorded for 50 different t_{run} values with 5 trials for each value. Every synapse is color-coded according to its preferred phase φ_n . The phases are marked in a blueish shade if they differ less than $\frac{1}{4}T$ from the mean phase of the postsynaptic spikes, otherwise they are marked in a reddish shade. The horizontal black line marks the initial weight w_{start} . Figure 18 already shows the dependence of synaptic weights after learning on the respective preferred phase very well, although the curves have a very noisy character. This is due to the fact that presynaptic spike times were generated anew for every run and would presumably be less pronounced if weight recording during a single simulation were implemented. However, this is technically difficult and the method used here is sufficient to show the weight dependence on the respective preferred phase.

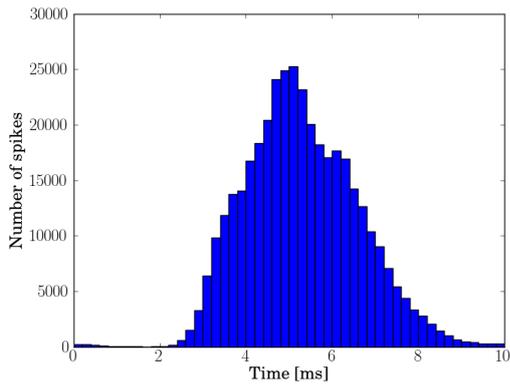
Figures 15(d) and 17(d) show the resultant weight distribution after 200 s of learning, for a single run and 20 accumulated runs, respectively. In both plots, the synapses are not distinctly parted into groups which either have the maximum or minimum synaptic weight as is the case in Gerstner et al. (1996) or the simulations in section 5.1 (see figures 6(d) and 8(d)). The weight of the synapses which survive learning is higher than w_{start} , but the



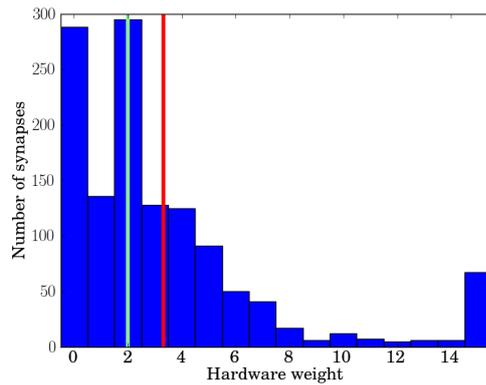
(a) Delays before learning.



(b) Delays of the synapses which survive learning.



(c) Postsynaptic spike times modulo T



(d) Synaptic weights after learning

Figure 17: Accumulated results of a 200s learning process, analogous to figure 15. Every run has a different distribution of the preferred phases φ_n and hence different presynaptic spike times. A mean number of 29 synapses have survived. The mean firing rate for all runs is 95 Hz, the mean vector strength is 0.87. As in the single run in figure 15, the mean weight increases during learning. The weights before learning are marked by the green line in figure (d), the mean weight after learning is marked red.

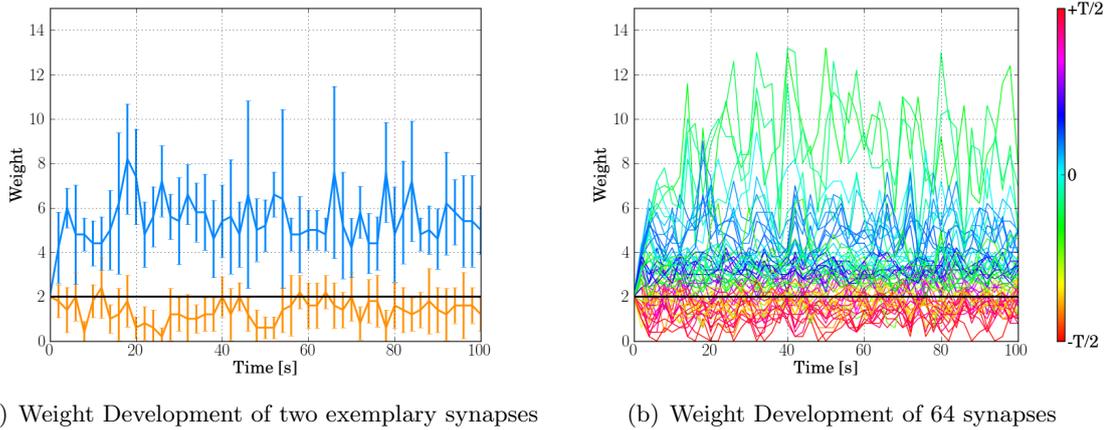


Figure 18: Weight Development for different runtimes. Figure (a) shows the weight development of one depressed and one potentiated synapse (orange and blue curves, respectively). The synaptic weight of the potentiated synapse shows more fluctuations for different runtimes than the weight of the depressed synapse. Figure (b) shows the same measurements for all 64 synapses. Synapses are color-coded according to their mean phase φ_n . For this, the mean spiking phase of the postsynaptic neuron was normalized to be zero. The color of every curve indicates the difference of a synapse's preferred phase from the phase of the postsynaptic neuron. For instance, synapses whose φ_n differ from zero by $\pm T/2$ are marked red. Synapses whose φ_n differs less than $\frac{1}{4}T$ are marked in a blue or green shade. The synaptic delay from the presynaptic neurons to the postsynaptic neuron was not taken into account for color-coding. This delay is not to be confused with the Gaussian distribution of the transmission delays, i.e., the mean phase distribution. The same color-coding as in figure (b) is used in figure (a). Figure (b) shows clearly that synapses with φ_n close to the mean phase of the postsynaptic spikes are potentiated while synapses which are out of phase of the postsynaptic spikes are depressed. The black line in both figures marks the initial weights w_{start} .

synapses are not saturated at w_{max} . This is the case for most simulation runs and can not be improved by a longer learning time. Figure 18 illustrates this since the weight distribution does not change significantly for larger run times. For this reason, only runtimes up to 100 s have been plotted in figure 18 to provide a better overview.

This weight distribution is presumably a result of the limited learning window and the too high maximum hardware weight and can be understood as follows. If the learning window is chosen smaller - in this case with the parameter set corresponding to the yellow curve in figure 13 - learning is more selective. Using this learning window has been found to result in more surviving synapses, but also a bimodal weight distribution as shown in figure 19(a) with two prominent groups of synapses - one with zero weight and one with the maximum weight. This corresponds to the model (Gerstner et al., 1996), but it also means a stronger total presynaptic stimulus than for the weight distributions in figures 15(d) and 17(d). The red line in 19(a), denoting the mean weight after learning, illustrates this. It is positioned at a hardware weight larger than 8, whereas the mean weight after learning in section 5.3.1 has been around 3. Consequently, the postsynaptic neuron receives a much higher total presynaptic stimulus and has a very high mean firing rate of about 250 Hz. At such a rate, the postsynaptic neuron not only fires once during every period of the input signal, which has a frequency of 100 Hz, but several times in quick successive bursts. This bursting behavior leads to a deviation of the postsynaptic neuron's spike times from its preferred phase. The spike times modulo T pictured in figure 19(b) then exhibit a broader or double peak structure, resulting in a lower vector strength, in this case $v = 0.70$.

This suggests that the learning window selected in section 5.2.3 has been chosen well, because it provides a compromise between good learning towards a high vector strength and a reasonable weight distribution. To achieve a weight distribution more similar to the software simulations *and* a high vector strength, smaller hardware weights resulting in a lower mean firing rate would be necessary. A narrower learning window could then be used to provide a distinct splitting of the synapses into synapses with w_{max} and synapses with zero weight.

Furthermore, a different maximum of the learning window could improve the weight distribution after learning. According to Gerstner et al. (1996), the time difference s^* for which the learning window assumes its maximum value is even more important for successful learning than the actual shape of the learning window. In order to achieve a maximal increase of synaptic weights for those synapses which are already strongest, the maximum of the learning window function $W(s)$ should be at the location at $s^* \approx \tau_s/2$ (Gerstner et al., 1996), τ_s being the synaptic time constant. This is the case in the original publication (see equation 1). It is possible to apply an offset to the hardware learning window in order to shift its maximum to a certain s^* . However, since the STDP curves of different synapses vary greatly on the chip (see figure 14), their maxima are likely to be very different, too. For this reason, s^* has not been shifted in this work.

5.3.3 ITD dependency

The preceding sections have shown that phase-locking works well on hardware. As a further step towards sound localization and in order to test the reliability of phase-locking, the

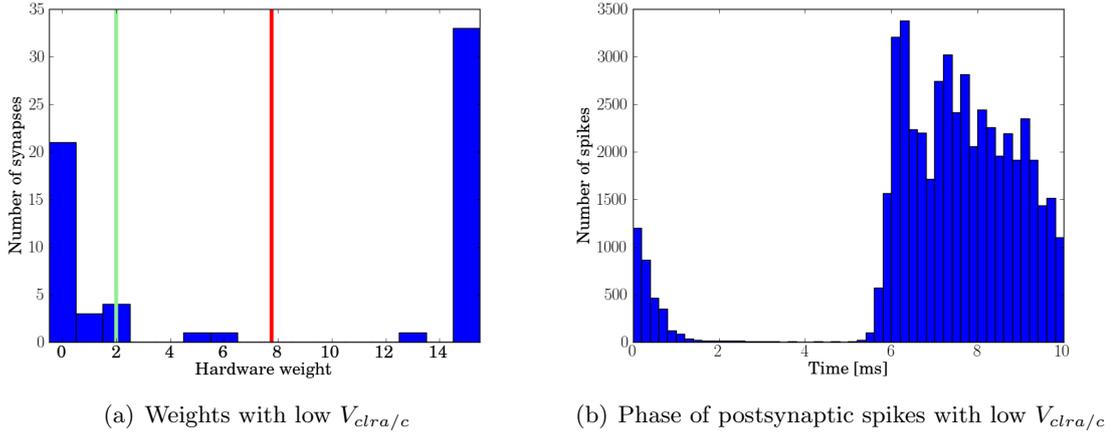


Figure 19: Synaptic weights and postsynaptic spikes for a modified learning window with STDP parameters corresponding to the yellow curve in figure 13, after 200s of learning. The bimodal weight distribution in figure (a) can be achieved with this learning window. This weight distribution leads to a much higher mean weight after learning (red line) compared to w_{start} (green line) with a correspondingly higher total presynaptic stimulus. Postsynaptic spikes plotted modulo T in figure (b) hence show a broad peak due to the high mean firing rate of 250 Hz. The vector strength is $v = 0.70$.

behavior of the trained postsynaptic neuron for stimulation with different interaural time differences is examined. As a neuron in the nucleus laminaris, the postsynaptic neuron is a coincidence detector (see section 2). It is driven optimally for the interaural time difference it was stimulated with during learning (Gerstner et al., 1996; Maass and Bishop, 2001). For other ITDs, the mean firing rate and the vector strength of the postsynaptic spikes are expected to be lower. Both the mean firing rate and the vector strength should have a minimum value for an ITD which differs by $T/2$ from the ITD the neuron was stimulated with during learning. In this study, the ITD is 0, since the neuron was trained to phase-locked spiking behavior for simultaneously arriving presynaptic spikes. The dependency of the postsynaptic neuron’s spiking behavior on the ITD is measured similarly to the method used in Gerstner et al. (1996). The postsynaptic neuron has been trained to $ITD = 0$ and synapses with similar delays modulo the period T have survived learning as in the previous subsections. STDP is now disabled and synaptic weights hence remain the same as after the learning process. For measuring the ITD dependency, the surviving synapses are then randomly separated into two equally large groups. Each group is considered to be transmitting the auditory signal from one ear. A phase offset $\Delta\varphi$ is applied to the preferred phase φ_n of every synapse in the second group. This network setup is emulated for $t_{run} = 200$ s and the mean firing rate and vector strength of the postsynaptic spikes are recorded. This is repeated for different values $0 \leq \Delta\varphi \leq T$, resulting in the curves shown in figure 20. The data plotted here was recorded using the weight and mean phase distribution of the single exemplary run depicted in figure 15, but with newly generated presynaptic spike times. The red curve in figure 20 shows the mean firing rate, the blue curve the vector strength for the respective ITD. Each data point is the mean of three runs. The recorded ITD dependency reproduces the corresponding figures 3 a and b in Gerstner et al. (1996)

very well. A high vector strength of more than 0.9 can be achieved for values $\text{ITD} = 0$ or $\text{ITD} = T$, for which the mean phase distribution modulo T is the same. The vector strength reaches its minimum of about $v = 0.5$ for an ITD of $T/2$. This value is comparable to the vector strength without STDP which is discussed in the next subsection. For $\text{ITD} = T/2$, the postsynaptic neurons receives a maximally incoherent presynaptic stimulus and hence exhibits very poor phase-locking. As the ITD is increased towards T , phase-locking improves rapidly until it reaches the maximum value again at $\text{ITD} = T$. The ITD dependency of the mean firing rate is similar. As expected, the postsynaptic neuron exhibits the highest firing rate for coherently arriving signals from the respective ears. This enables the determination of the ITD and hence the localization of the sound source in further processing steps as mentioned in subsection 2.1. The minimum firing rate is shifted slightly towards larger ITD values in the recorded data. This could be a point for further investigation. Nevertheless, the data plotted in figure 20 shows that the learning process has led to a reliable coincidence detection by the postsynaptic neuron. Thus, the network is indeed capable of resolving small phase differences.

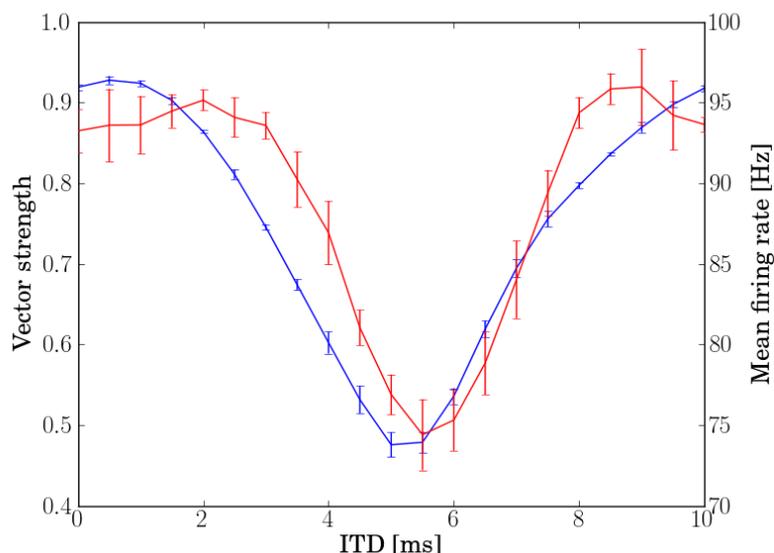


Figure 20: Dependency of the mean firing rate (red) and the vector strength (blue) of the postsynaptic spikes on the interaural time difference.

5.3.4 Measurements without learning

Control experiments are done to determine what happens for the same network and presynaptic input without learning. The postsynaptic spikes are hence recorded for the same time without STDP enabled. The weight w_{start} of the 64 synapses is increased to 3.5 in order to have enough presynaptic input for a postsynaptic firing rate of around 100 Hz. This is done for better comparison to previous hardware emulations where the firing rate has been around 100 Hz as well (see figures 15 and 17). Figure 21 shows the postsynaptic spike times for 20 independent runs, again plotted modulo T . As in section 5.3.1, the mean phase distribution and presynaptic spike times have been generated anew for every run. The mean vector strength of these measurements is $\bar{v} = 0.47$. Figure 21 shows no distinguishable preferred

phase in the postsynaptic neuron's spikes as opposed to the accumulated results in figure 17(c). This confirms the statement in 2.2 that no phase information can be conveyed without an appropriate selection of synapses.

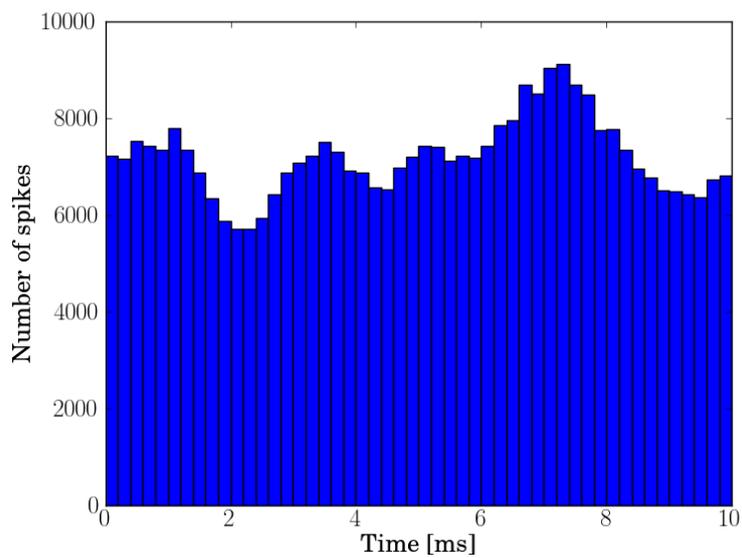


Figure 21: Postsynaptic spikes modulo the period T without learning. The mean vector strength is $\bar{v} = 0.47$.

6 Conclusion and outlook

Although many of the network parameters could not be set to the values according to Gerstner et al. (1996), good phase-locking can be achieved on neuromorphic hardware. With the presented adaptations, the postsynaptic neuron can be trained to emit phase-locked spikes with a mean vector strength of $\bar{v} = 0.87$. In Gerstner et al. (1996), a vector strength of 0.94 was reported for an input frequency of 2000 Hz, which is the frequency that corresponds to the temporally stretched setup used here. The resultant phase-locking on Spikey is hence nearly as good as in the original publication, even with all the hardware limitations. The essential basis for this is that the given network is well-suited for the hardware. Most importantly, it is very noise tolerant. Since possible sources of noise are already taken into account with the presynaptic jitter σ_φ and the broad delay distribution, additional noise due to the trial-to-trial variability of the hardware or thermal noise does not affect the network seriously. Learning may take longer due to slower and less precise weight updates, but a high temporal precision in the postsynaptic neuron’s spiking behavior can be achieved nevertheless as shown in sections 5.3.1 and 5.3.3. Phase-locking works reliably enough on hardware to show an ITD dependency similar to the results in Gerstner et al. (1996). The postsynaptic neuron exhibits maximum vector strength and firing rate for the ITD it was stimulated with during learning. For an ITD differing by $T/2$ from the stimulation ITD, it the neuron exhibits minimum vector strength and firing rate. It could thus be trained to be a temporally precise coincidence detector.

Software simulations with a model of the hardware synapses have been found to provide a good prediction for the hardware performance. Both software simulations and hardware emulations yield a mean vector strength of $\bar{v} = 0.87$. The burn-in time, by which most synaptic weights have developed towards their value after learning, is around 150s for both models. Also, similar delay distributions of the surviving synapses have been recorded. The weight distribution after learning differs significantly, however. On hardware, the synaptic weight distribution after learning does not show the characteristic bimodal distribution of synapses with either minimum or maximum weight as shown in Gerstner et al. (1996) and the software simulations. This is due to the fact that the maximum hardware weight is limited, whereas the maximum weight in the software model can be set arbitrarily. For the same reason, the mean firing rate of 89 Hz is larger on hardware than the measured 62 Hz for the software simulation.

The weight distribution after learning could be improved by calibrating the hardware synapse drivers to lower synaptic weights. With a weaker total presynaptic stimulus, a smaller learning window similar to the one used in 5.3.2 could then be used to obtain a distinctly bimodal weight distribution after learning. A further advantage of the synapse driver calibration would be to have equally strong synapses. Since synaptic strength has been found to vary greatly on the chip (figure 12), learning might be dominated by a few very strong synapses. This might have a distorting effect on learning if these strong synapses are out of phase with compared to the postsynaptic spikes. A calibration of the synapse drivers towards the same synaptic weight could prevent this.

Parameter adaptations for which time was limited in this thesis provide further possibilities for improvement. A detailed investigation of the plateau region in the learning windows in

section 5.2.3 is necessary to determine the shape of the learning window and τ_{STDP} accurately. If a more homogenous distribution of the learning windows of the involved synapses could be achieved, less synapses would need to be blacklisted, resulting in a higher weight update frequency. Also, a measurement of the EPSPs in high conductance state of the neuron and by means of spike-triggered averaging, as it is presented in Brüderle (2009), could be done. This would allow a more accurate determination of the EPSP width and area than the simple setup used in section 5.2.2. Building upon the method described in section 5.2.1, the existing standard calibration of the membrane time constant developed in Brüderle (2009) could be extended to include the I_{leak} dependency of τ_m . As a further improvement of the calibration, an automatic measurement of the refractory period could be included instead of the current manual measurement. Where no direct setting of the parameters is possible, the 10^4 speed-up of Spikey could then be utilized for parameter sweeps in order to determine optimum settings.

Given the already good phase-locking on hardware, a setup including further processing steps towards sound localization could be built (see also section 2.1). For this, a population of postsynaptic neurons would be needed, which have all been trained to exhibit their maximum firing rate for a different ITD. The information which neurons have the highest firing rate could then be processed in order to calculate the ITD and corresponding azimuthal location of the sound. A setup like this could be used, for instance, for a roboter which can detect the location of ultrasonic sounds. In summary, if some further steps are taken, the given setup may provide interesting biological applications which work reliably on the neuromorphic Spikey chip.

References

- D. Brüderle. *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System*. PhD thesis, 2009.
- W. Gerstner and W. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- W. Gerstner, R. Kempter, J.L. van Hemmen, H. Wagner, et al. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78, 1996.
- J. M. Goldberg and P. B. Brown. Response of binaural neurons of dog superior olivary complex to dichotic tonal stimuli: Some physiological mechanisms of sound localization. *Journal of Neurophysiology*, 32:613–636, 1969.
- W. Maass and C.M. Bishop. *Pulsed neural networks*. MIT press, 2001.
- A. Morrison, M. Diesmann, and W. Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics*, 98:459–478, 2008.
- NEST. A simulator for spiking neural network models – website. <http://www.nest-initiative.org>, 2008.
- T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier. Is a 4-bit synaptic weight resolution enough? - Constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Frontiers in Neuromorphic Engineering*, 6, 2012.
- PyNN. A Python package for simulator-independent specification of neuronal network models – website. <http://www.neuralensemble.org/PyNN>, 2008.
- J. Schemmel, A. Grübl, K. Meier, and E. Muller. Implementing synaptic plasticity in a VLSI spiking neural network model. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*. IEEE Press, 2006.
- A. Scherzer. Phase-locking mit feuernenden neuronalen Netzen, 2012.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Unterschrift Verfasser

Ort, Datum