Faculty of Physics and Astronomy

University of Heidelberg

Diploma thesis

in Physics

submitted by

Paul Müller

born in Kujbyshew, Russia

Mai 2011

Distortions of Neural Network Models Induced by Their Emulation on Neuromorphic Hardware Devices

Neuromorphic hardware represents the integration of neuronal computation paradigms with modern techniques of electronic hardware design. A general-purpose neuromorphic hardware device that implements a number of adjustable neuron circuits with configurable interconnections offers a powerful alternative to software simulation as substrate for neuroscientific modeling. This thesis investigates the distortions that arise from inevitable limitations and imperfections of a neuromorphic VLSI device and their influence on the behavior of neural network models. As variations during the manufacturing process lead to a spread of synaptic strengths on the neuromorphic device, a rate-based method is developed for measuring this effect. These results are later used to investigate the influence of synapse strength spread, incomplete realization of synaptic connections and absence of synaptic delays on models of a synfire chain and a self-sustaining thalamocortical network. The possibility of compensation of these effects by changing model parameters is examined, establishing the limits and feasibility of such methods.

Verzerrungen von Netzwerkmodellen aufgrund ihrer Emulation auf neuromorphen Hardwaresystemen

Neuromorphe Hardware stellt die Integration von Konzepten neuronaler Informationsverarbeitung mit modernen Techniken des Designs elektronischer Hardware dar. Eine universelle neuromorphe Hardware mit einstellbaren Neuronen-Schaltkreisen und konfigurierbaren Verbindungen bietet eine leistungsfähige Alternative zur Software-Simulation als Mittel neurowissenschaftlicher Forschung. In dieser Arbeit werden Störungen untersucht, die durch unvermeidliche Einschränkungen und Unvollkommenheiten eines neuromorphen VLSI-Systems entstehen, und deren Einfluss auf das Verhalten neuronaler Netzwerkmodelle analysiert. Da Abweichungen während des Fertigungsprozesses zu einer Streuung von Synapsenstärken auf dem neuromorphen System führen, wird eine Ratenbasierte Methode entwickelt, um diesen Effekt zu messen. Die Resultate dieser Messung werden benutzt, um zu untersuchen, welchen Auswirkungen eine Streuung von Synapsenstärken, eine unvollständige Realisierung synaptischer Verbindungen oder die Abwesenheit von synaptischen Verzögerungen auf das Verhalten eines Synfire-Chain-Modells und eines selbsterhaltenden thalamo-kortikalen Modells haben. Möglichkeiten, diese Verhaltensänderungen durch Änderung von Modellparametern zu kompensieren, werden analysiert sowie Machbarkeit und Grenzen eines solchen Vorgehens festgehalten.

Contents

Introduction

Physics can be described as the strive for the understanding of the fundamental laws of nature that govern the universe and the incredible amount of phenomena that emerge from these laws, ultimately creating the whole universe from clouds of gas to stars and galaxies, from a single molecule to living organisms. All of this is, ultimately, the result of the building blocks of the universe interacting with each other according to those fundamental laws. It was only a question of time until the attention of scientists was drawn to a highly complex mechanism that is present in a huge amount of different realizations and that shapes the existence on earth in a fundamental way. This mechanism is the brain, an information processing mechanism that contributed to survival such an enormous way that it has been shaped by evolution in a great variety of shapes.

The understanding of the brain, which is itself a composition of interacting basic building blocks, the neurons, which interact according to a set of given properties, should greatly benefit from the expertise and analysis methods that are established in the field of physics for similarly composite, but more basic systems.

Naturally, any investigation of something as involved as the brain can only be performed in a large scale, interdisciplinary effort. The necessary steps to discover and understand the functionality of single neurons, their interconnection and interaction and understand the emergence of functionality for many different parts of the brain requires specialists to perform a wide variety of research ranging from anatomy, physiology and behavioral studies to mathematical models of computation paradigms.

An important tool to the understanding of the function of individual brain areas is the simulation of neural networks using computer hardware, modeling individual neuronal behavior and their interconnection as precise as possible from available biological data, and using the results to obtain insights to further the neuroscientific research. In cases where the emergent behavior is of a very complex nature that escapes direct experimental access, this is one of the few viable options. This approach is especially lucrative due to the readily available computing power in numerous data centers, that enables the simulation even of very large networks, albeit the simulation of e.g. a full-size human brain with any degree of realism still remains unfeasible.

Instead of waiting for Moore's law to take care of missing computing power, attempts are made to create fast, scalable and energy efficient alternatives to conventional simulation. One attempt is called *neuromorphic hardware*: Instead of using a digital computer to run software that solves differential equations that describe the behaviour of a set of electric circuits which behave similarly to a network of neurons, one uses available methods of circuit design to *build* the set of electric circuits. In short, one replaces the *simulation* of physical processes by an *emulation* of physical processes by other physical processes. For example, as a very basic neuron can be viewed as a capacitance in parallel with a number of resistors with eventually varying conductance, and resistance can be built as an electric circuit.

By choosing appropriate values for the electrical circuit components, the emulation can be performed faster than the operation of real neurons, making it possible to conduct experiments

at a high speed. This speed increase does not depend on the number of implemented neurons, which is a huge advantage to software simulation that slows down at least linearly with the number of simulated neurons.

The disadvantage of this approach is the loss of flexibility that is inherent to the traditional approach of digital computation. Thus, the most important idea is to build a configurable device that still allows the emulation of neural behavior but additionally makes it possible to configure a large number of different neuronal networks on the same device, thus creating a universal modeling tool. This is the idea behind the FACETS and BrainScaleS hardware systems.

The fact that the hardware system is still in its infancy and that entirely new concepts need to be drawn from scratch, much work needs to be done before the concept can reach its full potential. In the meantime, the limitations of the hardware system need to be investigated and understood. These limitations arise from a fixed number of implemented hardware components, from design decisions regarding adjustable parameter ranges and number of possible interconnections between neuron circuits, to noise that is inherent to analog circuits and production variability that affects analog circuits as well. It is extremely important to analyze their effects and find ways around them, within the configuration space of the hardware. For evident reasons, this analysis needs to include neuroscientifically relevant neural network models to establish which limitations are most important. Conversely it can be established which model parameters are related to the functionality of the network, enabling future modeling to make use of the strength of the hardware systems while evading its weak points.

The concern of the thesis is particularly the task of establishing the influence of the limitations on neural network models. The concept is to use a given set of distortions that are known to be introduced by hardware emulation, measure their extent and investigate their influence on the neural models. The severity of distortions is seen by how well a model change can restore the original behavior.

Thesis Outline

In Chapter 1, the FACETS/BrainScaleS hardware systems are described. Chapter 2 describes the FACETS Demonstrator project and the included neural network models used as benchmarks. Chapter 3 concerns a detailed description of the investigated hardware distortions and the way they are modeled during the analysis. Chapter 4 details the measurement of synaptic strength variability on the chip-based neuromorphic hardware system. Chapter 5 contains the investigation of the distortion influence on two of the Demonstrator benchmark models and the results of distortion compensation by change of model parameters.

1. The FACETS Hardware Environment

In this chapter, the hardware devices that are relevant for this thesis are described. The chip-based neuromorphic device described in 1.1 is used in chapter 4 to investigate variations of synaptic strengths. The analysis in chapter 5 is conducted with respect to the wafer-based neuromorphic system that is described in 1.2.

Additionally to the cited sources, the information presented in the following sections is based on personal communication with Daniel Brüderle and Bernhard Vogginger

1.1. Chip-Based Neuromorphic System

The chip-based neuromorphic hardware system is based on the mixed-signal Application-Specific Integrated Circuit (ASIC) *Spikey*. The *Spikey* chip was developed in the Electronic Vision(s) group and was available in its 4th development version during the course of this thesis.

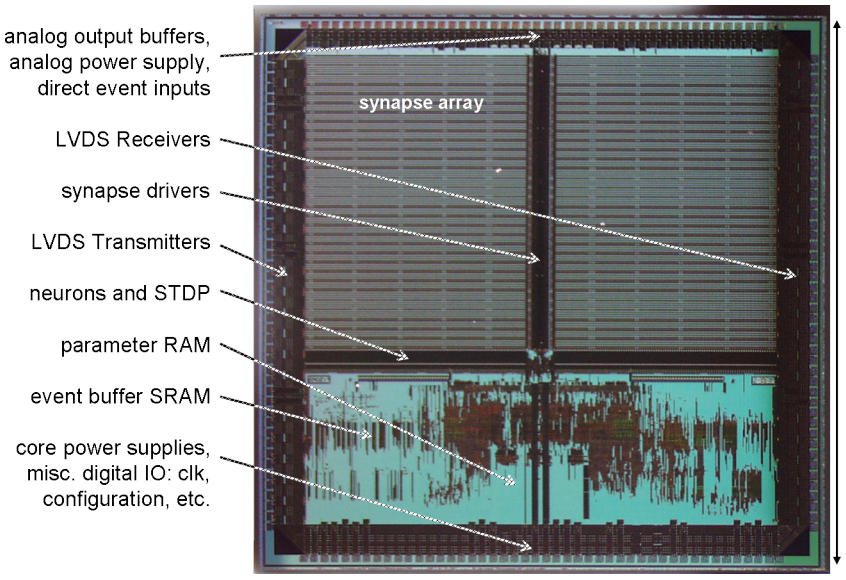


Figure 1.1.: Photograph of a Spikey chip. Image taken from *Schemmel et al.* [2007]

It implements 384 analog neuron circuits that emulate a *conductance-based leaky integrate-and-fire* (LIF) point neuron [*Gerstner and Kistler*, 2002, chapter 4.1] with exponentially decaying synaptic conductances. The behavior is given by the following equations:

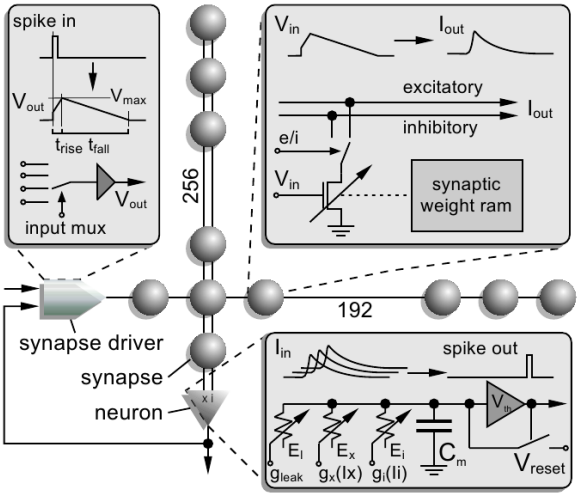


Figure 1.2.: Image taken from *Schemmel et al.* [2007]

$$C_{\mathrm{m}}\frac{dV}{dt} = -g_{\mathrm{l}}(V - V_{\mathrm{rest}}) - \sum_j g_j(t)(V - E_{\mathrm{rev,E}}) - \sum_k g_k(t)(V - E_{\mathrm{rev,I}}) \tag{1.1}$$

$$V \leftarrow V_{\mathrm{reset}} \text{ when } V > V_{\mathrm{thresh}} \tag{1.2}$$

$$g_j(t) = w_j \cdot \Theta(t - t_j) \cdot \exp\left(\frac{-(t - t_j)}{\tau_{\mathrm{syn}}}\right) \tag{1.3}$$

with Θ being the heaviside step function.

V is the neuron's membrane potential, C_{m} the membrane capacity, g_{l} the membrane's leak conductance. The membrane potential returns exponentially to V_{rest} when synaptic input is absent. Synaptic input is given in the two sum terms, the first for excitatory and for inhibitory synapses. Each firing synapse connects the membrane to a voltage of $E_{\mathrm{rev,E}}$ for excitatory and $E_{\mathrm{rev,I}}$ for inhibitory synapses via its time-dependent synaptic conductance g_j (resp. g_k). When V reaches a threshold value V_{thresh}, it is set to the reset value V_{reset} and clamped there for a refractory period τ_{refrac}.

The synaptic conductance of the i-th stimulus is zero before the spike time t_i, at which the conductance jumps to a value w_i (the synaptic weight) and decreases exponentially with the time constant τ_{syn}. In the case of *Spikey*, two synaptic time constants can be configured, one for excitatory and one for inhibitory synapses.

Because of the small scale of the neuron circuits, the possible values for the membrane time constant $\tau_{\mathrm{m}} = C_{\mathrm{m}}/g_{\mathrm{l}}$ are much smaller than usual biological values. Thus, all neuron parameter ranges were chosen such that the emulation runs at a *speedup* of 10^4, i.e. an experiment that would take ten seconds in real-time would be emulated in one millisecond on the chip. This fact makes it necessary to distinguish between *hardware time* and *biological time*, the former denoting the time that the hardware system needs to emulate a given setup and the latter being the time that has been translated back with the *speedup* factor.

Synaptic input is realized via two components: The synapse circuits, located in the synapse array, and the synapse drivers, that are located in between the two synapse arrays. (see

Figure 1.1) A synapse driver can receive spikes from an on-chip neuron, allowing recurrent connections, or from an outside source, allowing external stimulation. When a spike has to be emitted by the given source, the synapse driver generates a voltage course with a linear rising and falling edge (see Figure 1.2). The shape of this voltage course can be modified using hardware parameters. It is then injected into the row of synapses that are associated with the driver. The piecewise linear voltage is converted into an exponentially rising and falling current, respectively. (For details, see *Schemmel et al.* [2007]) Each synapse that is connected to the driver and a neuron, and is configured with a non-zero weight, sends the current course to the neuron. There, a circuit interprets the current course as a conductance between the membrane voltage and the corresponding (excitatory or inhibitory) reversal potential.

The mechanism of voltage course generation in the synapse driver implies that synaptic conductances that originate in the same driver are not additive; each incoming spike induces a new time course generation. In other words, the system behaves as a saturating synapse.

The behavior of the synapse circuits becomes important for the analysis presented in chapter 5.1.

In addition to the conductance dynamics mentioned above, synaptic plasticity mechanisms are implemented on the chip: *Spike Timing Dependent Plasticity* (STDP) (*Bi and Poo* [1997]) and *Short Term Plasticity* (STP) (*Markram et al.* [1998], (*Schemmel et al.* [2007]). As these mechanisms are not of relevance for the scope of the thesis, they will not be detailed further at this pont.

For further information about the *Spikey* chip, see e.g. *Grübl* [2007].

1.2. Wafer-Based Neuromorphic System

The next-generation system within the FACETS and BrainScaleS projects is a wafer-scale neuromorphic hardware system. While the chip-based neuromorphic system is scalable via connection of several *Spikey*-based boards, the wafer-based system will allow an efficient integration of neuromophic modules by leaving the production wafer intact and connecting the modules in a post-processing step. (*Schemmel et al.* [2008], *Schemmel et al.* [2010], *Jeltsch* [2010])

1.2.1. Neuron and Synapse Functionality

The basic module on a wafer is called High Input Count Analog Neural Network (*HICANN*). Each HICANN incorporates 512 neuron circuits with maximally 224 synapses connecting to each neuron. Up to 64 neighboring circuits can be interconnected to increase the number of afferent inputs. This imposes a hard upper limit on either the total number of available neurons or the number of input synapses per neuron.

The HICANN module implements the *Adaptive Exponential integrate-and-fire* (AdEx) neuron model (*Brette and Gerstner* [2005]). It is defined by the following equations:

$$C_m \frac{dV}{dt} = -g_L(V - V_{\text{rest}}) + g_L \Delta \exp\left(\frac{V - V_{\text{thresh}}}{\Delta}\right) - w - I_{\text{syn}} \tag{1.4}$$

$$\tau_w \frac{dw}{dt} = [a(V - V_{\text{rest}}) - w] \tag{1.5}$$

$$I_{\text{syn}} = \sum_j g_j(t)(V - E_{\text{rev,E}}) + \sum_k g_k(t)(V - E_{\text{rev,I}}) \tag{1.6}$$

$$\text{when } V = V_{\text{cutoff}}: \tag{1.7}$$
$$V \leftarrow V_{\text{reset}} \tag{1.8}$$
$$w \leftarrow w + b \tag{1.9}$$

The essential differences to the LIF model (equations 1.1 to 1.3) are the addition of an adaptation variable w, which allows for a wide variety of firing patterns that are observed in nature (*Touboul and Brette* [2008]), and the introduction of an exponential term that affects the membrane dynamics at voltages approximately equal to, and greater than the threshold value V_{thresh}. *Millner et al.* [2010] shows that the firing patterns were successfully emulated in hardware.

The variable Δ determines how abruptly the exponential term becomes relevant in the neuronal dynamics.

When V reaches V_{cutoff}, the membrane potential is reset and clamped to V_{reset} for a refractory period τ_{refrac}. Simultaneously, the adaptation variable w is increased by an offset b. The behavior of the synaptic conductance terms is identical to that of equation 1.3.

The manner in which the synaptic time course is generated in the HICANN module differs from the one on the Spikey chip. Instead of generating a voltage time course in the synapse driver and transferring it to the neuron circuit via the synapses, the variation of the synaptic conductance is generated in each neuron circuit itself. The information about a spike that is sent out by the synapse is a square current pulse that only encodes the efficacy of the synapse *Schemmel et al.* [2008]. For technical reasons, this architecture is expected to be more robust in terms of synapse strength variations than the one used on the *Spikey* chip.

Just like on the single-chip system, synaptic STP and STDP are also incorporated. For further details on the implementation, the reader is referred to *Schemmel et al.* [2008] and *Schemmel et al.* [2010]

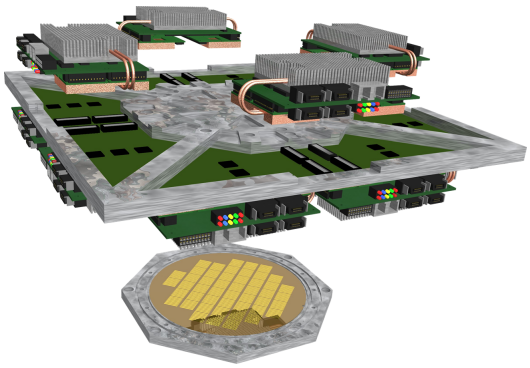


Figure 1.3.: Rendering of wafer-based hardware system. Image provided by Dan Husmann de Oliveira

1.2.2. Inter-, Intra-Wafer and External Communication

The communication between HICANNs that are located on the same wafer is realized via a grid of so-called *Layer 1* buses. Each HICANN module incorporates horizontal and vertical buses (represented as green arrows in figure 1.4) which enable the communication between neurons. When a neuron circuit emits a spike, it writes its address on a horizontal bus. At each HICANN border, repeaters are implemented which allow the transmission of spike events

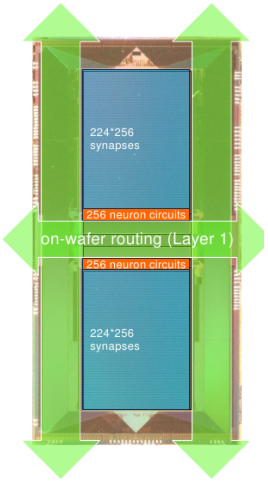


Figure 1.4.: HICANN module. Arrow overlays show the location of horizontal and vertical Layer 1 buses. The location of the synapse and neuron circuits is shown. Image provided by Electronic Vision(s) group.

to neighboring HICANNs. By appropriate switching between horizontal and vertical buses, a connection can be established to a vertical bus on the target HICANN. The vertical bus is switched to the target synapse driver that enables the forwarding of the spike information to the target neuron via the synapse array. Spikes distributed via the Layer 1 communication system reach their targets with a delay of less than 100 ns which corresponds to a synaptic delay smaller than 1 ms at a speedup of 10^4.

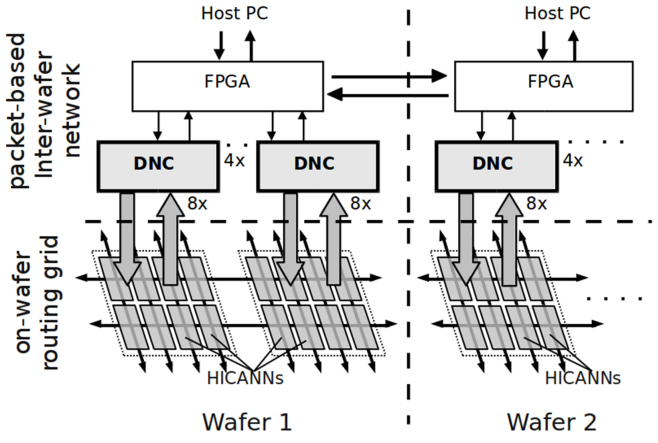


Figure 1.5.: Schematic showing the communication on a possible configuration of a wafer-based neuromorphic hardware system. Description is provided in text. Image provided by Christian Mayr.

The communication path between a HICANN module and an external component such as a HICANN on a different wafer or a host computer, including configuration and spike data, uses the *Layer 2* communication. A schematic is shown in Figure 1.5. Eight HICANNs are connected to a digital network chip (DNC) via the *DNC Interface*, a digital part of the

HICANN layout. Four DNCs are connected to an FPGA (Field Programmable Gate Array). The FPGA can be connected to a host computer or another FPGA.

A feature of the Layer 2 communication allows the realization of adjustable delays: Each neural event carries an id of its source neuron and its target Layer 1 bus as well as the time at which it should be delivered. This mechanism is implemented to counter variable transmission times due to varying communication load. The time stamp is used to inject the spike message at the appropriate time into the Layer 1 bus. By adjusting this time stamp appropriately in the DNC or FPGA, a transmission delay can be introduced that is higher than the transmission latency. Synaptic delays above 5 ms (biological time) between 2 neurons can be implemented with this feature by routing neural events from a HICANN via the Layer 2 network back to the wafer.

The availability of a communication path between two neurons is not necessarily given. The process by which neurons from an abstract network definition are assigned a concrete hardware neuron circuit (the *mapping*), and the algorithm that establishes the switching configuration between buses (*routing*) influence whether a desired synaptic connection can be established, because the number of available buses is limited. The most efficient mapping algorithm employed at this time is the *NForceCluster* algorithm that assembles neurons by similarity into groups that fit on a HICANN. These neurons are placed together on a HICANN module. For more details see *Wendt et al.* [2007].

1.3. PyNN

PyNN is an Application Programming Interface (API) that serves as a back-end-agnostic wrapper for a number of established simulators for spiking neural networks *Davison et al.* [2008]. It provides a means to describe neural networks in an abstract manner using the programming language *python*. Because it defines a standard for neuron types and units and hides simulator-specific details in implementation or representation of results by exposing a simple, but expressive API, neuroscientific experiments can be set up easily and executed on any of the supported back-ends only by changing the name of the desired neural simulator. This kind of abstraction is beneficial in several ways: The training effort of switching between simulator back-ends vanishes, providing the advantages of easy cross-checks of simulation results and allowing to exploit the strengths of each simulator in its own domain. By adding a new simulator to *PyNN*, the previously implemented network definitions and analysis programs become instantly available without a need of individual transfer of source code. The list of already established back-ends for *PyNN* is shown in figure 1.6.

For these reasons, the integration of the FACETS/BrainScaleS neuromorphic project as available *PyNN* back-ends is highly beneficial to both, the project itself and to the neuroscience community. The latter benefits, because the optimization focus of the hardware system differs significantly from pure software simulators: An exact, binary reproducibility of results is abandoned for a high simulation speedup that is independent of the network model complexity, up to the maximal number of available neurons on the hardware system.

For developers of a hardware system, the easy cross-check with an established simulator highly simplifies the tasks of assessing the performance, validating functionality of and, ultimately, finding possible error causes in the hardware system.

Currently, a *PyNN* interface to the chip-based neuromorphic system is available and in use. An advanced implementation that includes both hardware systems and generalizes features

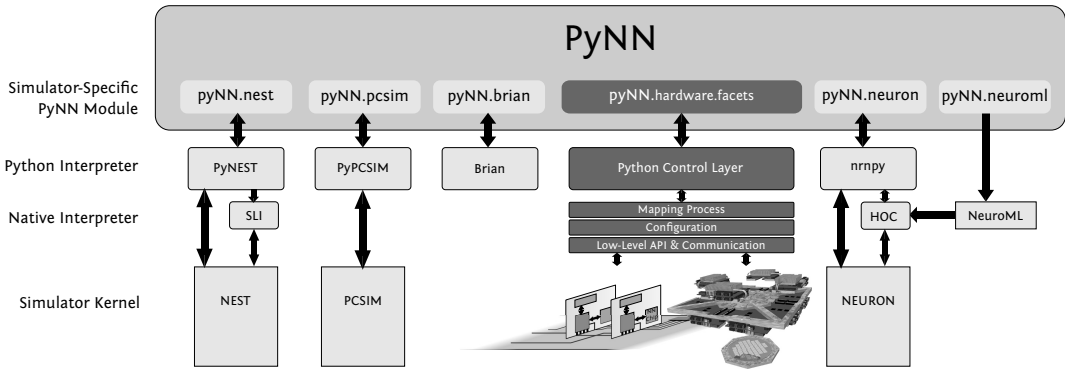


Figure 1.6.: Schematic of the modeling language PyNN with possible back-ends. Taken from *Brüderle et al.* [2011]

common to all neuromorphic hardware systems such as the mapping algorithms, is being actively developed.

For these reasons, all hardware and software experiments in the scope of this thesis were conducted using *PyNN*. Even though the benchmark models that are described in chapter 5 are only investigated in software because the wafer-based neuromorphic system was not complete during the course of this thesis, the implemented models are expected to be executable with minimal changes on any *PyNN*-compatible platform.

2. The FACETS Demonstrator

The FACETS Demonstrator project is an effort to showcase the functionality of the FACETS hardware systems and to provide a means to evaluate design decisions and anticipate possible problems during development. Furthermore it encompasses the development of the software infrastructure necessary for the operation of a neuromorphic hardware system, and the establishment of workflows concerning the setup and evaluation of analysis routines. Additionally, methods are being evolved that can be employed to compensate for hardware-specific effects. For this purpose, an executable system specification (ESS) has been created that allows to simulate several important aspects of the wafer scale hardware system, including the impact of mapping, routing and interneuron communication on the overall emulation performance. For a more detailed treatment of the FACETS/BrainScaleS ESS consult e.g. *Vogginger* [2010].

An essential part of the Demonstrator are benchmark models that serve as a basis for the aforementioned evaluation. For complex systems like the FACETS hardware devices, a large number of trade-offs regarding e.g. communication bandwidth, emulation speed, homogeneous behaviour of components and component number need to be made. The effects of such changes on the overall quality as a neuromorphic modelling tool are often hard to predict.

The benchmark models constitute one approach to this problem. A set of high-level network models with a well-defined behaviour is used to study the influence of expected hardware distortions. Each model is adapted from a published study, which establishes its neuroscientific and biological relevance. As such, it provides typical connection densities and patterns, neuron firing times, neuron parameters etc.

In particular, the following benchmark models were provided by FACETS project partners.

2.1. Synfire Chain with Feed-Forward Inhibition

The synfire chain model was provided by *L'Institut de Neurosciences Cognitives de la Méediterranèe – INCM, Marseille, France* in cooperation with *Albert-Ludwigs-Universität Freiburg – ALUF, Freiburg, Germany.* The corresponding publication is *Kremkow et al.* [2010].

A synfire chain is a set of groups of neurons in which members of one group connect to neurons in the group's successor. It is a theoretical tool that has been employed to study signal propagation, memory capacity and computational properties of neural networks. *Kremkow et al.* [2010] investigates the effect of feed-forward inhibition on signal propagation along a synfire chain. Each group contains excitatory and inhibitory neurons that are both stimulated by the preceding group's excitatory population; the inhibitory neurons only connect locally, influencing the response of the group. The model is described in full detail in 5.1.2.

The importance of this model as part of the Demonstrator stems from its highly synchronised firing pattern that may be affected by hardware bandwidth limitations while a relatively simple interconnection scheme allows to test the mapping and routing algorithms. The dependence of the model's functionality on delays can provide insights about impact of limited delays on the wafer based hardware system.

2.2. Self-Sustaining Cortical Activity with Asynchronous Irregular Firing Patterns

The network models were provided by the *Integrative and Computational Neuroscience Unit – UNIC of the Centre national de la recherche scientifique – CNRS, Gif-sur-Yvette, France.* The corresponding publication is *Destexhe* [2009].

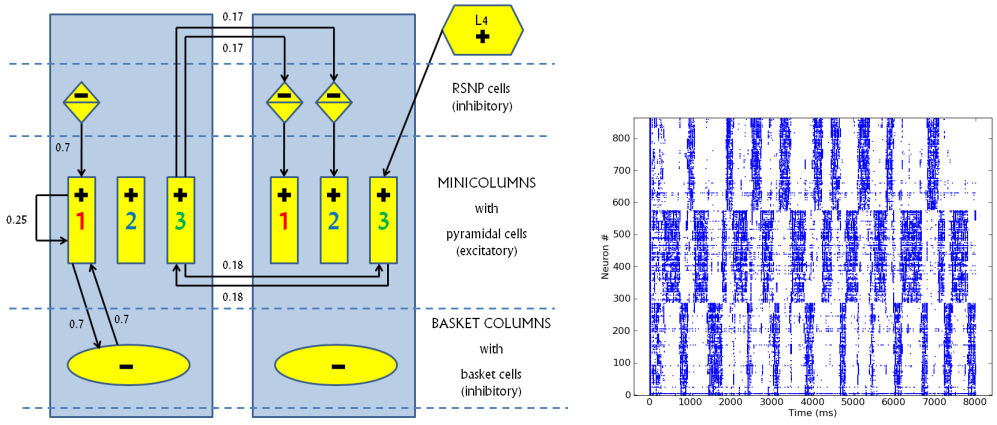
The networks employ the adaptive exponential neuron model to simulate neuron firing patterns that occur in the mammalian cortex and thalamus. With these, *Destexhe* [2009] shows that self-sustained asynchronous and irregular activity can be facilitated by the presence of rebound bursting neurons. The level of intrinsic adaptation is linked to the occurrence of Up and Down states.

The benefit of these models for the Demonstrator is their reliance on the adaptive exponential neuron model that is implemented in the HICANN module. Also, their random connectivity provides the opposite to the structure of the synfire chain model in terms of mapping simplicity.

The models are described in full detail in 5.2.1.

2.3. Layer 2/3 Attractor Memory

The Layer 2/3 Attractor Memory model was provided by *Kungliga Tekniska Högskolan - KTH, Stockholm, Sweden.* The corresponding publication is *Lundqvist et al.* [2006].



(a) **(b)**

Figure 2.1.: (a) Layer 2/3 Attractor Memory schematic. Numbers on arrows denote connection densities. Further description in text. (b) Raster plot of a NEST simulation of a network with 9 hypercolumns and 3 minicolumns per hypercolumn. The spike trains are sorted by minicolumn number. Pictures provided by *Mihai Petrovici.*

The mammalian cortex is organised in 6 distinct layers that are distinguished by their specific neuron types as well as both their intrinsic and mutual connectivity patterns.

The cortex is divided into hypercolumns, which run orthogonally to the surface and span all six cortical layers. The definition of a hypercolumn is based on its constituent neurons having nearly identical receptive fields ((*Mountcastle* [1979], *Buxhoeveden and Casanova* [2002], *Hubel and Wiesel* [1977])), which, however, makes a clear distinction between neighboring

hypercolumns somewhat difficult (*Tsunoda et al.* [2001]). According to an established hypothesis, each hypercolumn consists of 50 - 100 minicolumns, each comprising around 80 neurons. (*Buxhoeveden and Casanova* [2002])

In *Lundqvist et al.* [2006] it is shown that a hypercolumnar/minicolumnar structure with experiment-based neuron models and connectivity can act as an associative memory. The observed attractor dynamics shows traits such as pattern completion and pattern rivalry.

The network architecture is shown in Figure 2.1 (a). It is comprised of a set of hypercolumns (blue rectangle) each of which contains an equal number of minicolumns (yellow rectangles) and a basket cell population (yellow ellipses). Each minicolumn has an assigned group of regular spiking non pyramidal (RSNP) cells (yellow rhombus) which have an inhibitory projection onto their corresponding minicolumn. The minicolumns are grouped in so-called *orthogonal patterns*, each pattern having one minicolumn in each hypercolumn, with no minicolumns being shared among patterns.

Each minicolumn has excitatory connections to other minicolumns in the same pattern and to the RSNP cells of all remaining patterns, except in its own hypercolumn. This causes an activity increase in one minicolumn to excite its pattern while simultaneously inhibiting the other pattens. Additionally, each minicolumn has local recurrent excitatory connections.

All minicolumns also have excitatory projections to the basket columns of their hypercolumn; the basket column equally has inhibitory projections to all minicolumns in its hypercolumn. This can be regarded as a soft winner-take-all (WTA) module, which ensures a balancing of activity in each hypercolumn.

The network is stimulated by external input from layer 4 (yellow hexagon) and by diffuse input from other cortical layers.

Figure 2.1 (b) shows a typical spiking pattern of the network. All minicolumns in a pattern are active at the same time. This activity switches between patterns after several milliseconds.

This model is the most complex of the Demonstrator benchmark models. Its high-level functionality relies on the interplay of many populations. On the other hand, its self-regulatory setup may automatically compensate for hardware imperfections. While this analysis is not part of this thesis, a description of the model has been included for the sake of completeness. A detailed analysis can be found in *Brüderle et al.* [2011].

3. Modeling Hardware-Induced Distortions

While offering crucial advantages over conventional software simulators, the FACETS wafer-scale neuromorphic back-end does suffer from certain specific limitations. It is essential for these to be understood prior to the actual use of the hardware for modeling purposes. On one hand, a detailed study of hardware-induced distortions allows the development of appropriate corrective measures, such as calibration routines or even the re-design of (individual parts of) the hardware itself. On the other hand, a complete understanding also implies the study of their effects on a sufficiently diverse range of benchmark neural network models and, where possible, the design of suitable and preferably universal compensation methods.

Hardware-induced model distortions stem from the physically limited nature of the hardware itself, from its design and from the manufacturing process, but also from the software components of the operation workflow. In the following, a shortlist of the most relevant limitations is given, with an enhanced focus on those effects which were subject of this thesis. This serves as an introduction and a preparation for the following sections, which concern a detailed study of synaptic weight jitter (Chapter 4) and the development of the abovementioned compensation techniques for the FACETS Demonstrator benchmark models (Chapter 5).

3.1. Causes of Distortions

The highly complex set of hardware and software components that make up the neuromorphic hardware system contains many points at which limitations have to be accepted or where physical constraints introduce unavoidable distortions. Design decisions limit many factors of the hardware devices, the most important being the number of realized neuron and synapse circuits, the available communication bandwidth between neurons and to external devices and the available parameter ranges of many components. Physical constraints inevitably introduce cross-talk and noise and impose delays on the communication between any two parts of the system. Process variations during manufacturing introduce fixed pattern noise and result in variations of behavior in components that were designed to be identical. As mentioned in section 1.2.2, sophisticated algorithms have to be employed to perform the highly nontrivial task of mapping a given neural network onto the hardware in a way that ensures the highest possible fidelity to the original connectivity pattern. Due to the complex nature of the task, the choice of the concrete algorithm may significantly influence the actually realizable synaptic connections, in addition to any hard limit imposed by available communication resources.

3.2. Modeling Distortions

Since the waferscale device itself has not been available within the timeframe of this thesis, most of the effects mentioned above and discussed later on have not been available for direct measurement. However, realistic estimates have been used, which have either been measured

on the Spikey prototype (i.e. synaptic weight jitter, Chapter 4) or have been inferred from the system design (e.g. delays). When neither was possible or feasible, the investigated range was chosen large enough to ensure that it encompasses all realistic use cases (e.g. synaptic loss).

The distortion estimates were implemented in software simulations, as detailed below. Simulating these effects rather than measuring them directly on the hardware actually offers a significant advantage: while on the hardware all distortion mechanisms act simultaneously, making it difficult to analyze their individual effects, in software they can be investigated independently.

3.2.1. Synapse Loss

As mentioned earlier, the number of realizable synapses depends not only on hard constraints of the hardware design but also on the interchangeable and configurable mapping algorithm. Since the study presented in this thesis is intended to be independent of any actual mapping implementation, the loss of synapses has been modeled as being independent of any properties of each synapse, meaning that a *synapse loss* value of 10% is considered homogeneous and will result in every synapse being realized with a probability of 90%.

3.2.2. Spacial Synaptic Weight Jitter

Inevitable irregularities during production lead to deviations in components that were designed to be identical. For the individual synaptic circuits, as well as for the synapse drivers, their efficacy (i.e. their impact on their target neuron) differs from the ideal value. While this effect can be significantly reduced by calibration in case of the synapse drivers, individual synapses have no parameters which can be externally tuned.

Additionally, the digitalization of synaptic weights during the translation from the original model to a hardware representation changes their value by an amount that is determined by the resolution of synaptic weights, the maximal weight of the synapse and the desired weight in the model.

These deviations are implemented in software simulations by setting each synaptic weight that originally would have had a value of μ to a value that is chosen from a Gaussian distribution with mean μ and standard deviation $\mu \cdot j$. If the new value is smaller than zero, it is set to zero. This is done because the *PyNN* interface does not allow negative weight values.

$$p(w\prime) = \mathcal{N}(\mu, (\mu \cdot j)^2) \tag{3.1}$$

$$w_{\text{new}} = \max(w\prime, 0) \tag{3.2}$$

The quantity j is referred to as *spacial synaptic weight jitter* or simply *weight jitter*.

3.2.3. Limited Availability of Delays

In this thesis, the focus lies on the investigation of pure on-wafer connection routing. Because the on-wafer delays are negligible (.1 ms biol. real time), this effect can be easily implemented in software simulations by setting all delays to 0. Delays generated by the Layer 2 routing require a more involved characterization, which has only recently been initiated (Unpublished results from Capo Caccia Cognitive Neuromorphic Engineering Workshop).

4. Investigation of Weight Jitter on the Chip-Based Neuromorphic System

4.1. Introduction

The central aim of this chapter is the quantification of the weight jitter on the chip-based neuromorphic system. For this purpose, several methods are considered and a value of jitter is measured on the hardware. The goal is to provide an estimate of the expected weight jitter that shall be used in chapters 5.2 and 5.2.2 and to investigate possible ways to quantify the effect of production disparities on neuron behavior. While the synapse circuits differ greatly between the chip-based and wafer-based neuromorphic systems, there are several reasons for conducting the presented research. Because it is expected that synaptic strength variation will be lower on the wafer-based system, as mentioned in section 1.2.1, the presented results are taken as an upper bound for future hardware generations. Second, experience with an actual neuromorphic hardware system was deemed absolutely necessary to gain expertise and intuition for its behavior.

Johannes Bill [*Bill*, 2008, Ch. IV.3] investigated the variation of synaptic effects on the membrane potential on the third version of the Spikey system. The method used the variability in the excitatory post-synaptic potential (EPSP) integral as a measure for synaptic variability, establishing a lower bound for the inherent variation of 10.8%. The used method enabled a reasonably fast measurement.

There are several reasons to consider a different approach. First of all, the method employed by *Bill* [2008] uses Spikey III specifics to speed up the acquisition of data, which are not valid for the current, fourth version and are certainly not generalizable to different hardware implementations. Second, it requires the recording of the voltage potential, which implies the use of an external oscilloscope. A purely spike based method is favorable because it allows easier automation and scalability, as spike data uses less bandwidth than voltage traces. Third, a spike based approach may be useful if the measurement can be taken in a regime of operation that is close to that of a complex neuroscientific experiment.

4.2. Considered Methods

A measurement method for synaptic weight jitter on neuromorphic hardware is considered feasible when it fulfills the following requirements:

speed As the methods are investigated with large-scale testing in mind, execution speed is essential.

expressiveness The measured quantity can be easily translated to a value of the weight jitter quantity defined in 3.2.2.

precision The measurement should be exact.

realistic setup Because different effects can dominate for different experiment setups, it is necessary to conduct a measurement in a common use case. Especially load on the neuron by synaptic background activity is to be considered carefully.

automation The application of any jitter measurement method to a neuromorphic device with several million synapses needs to be fully automatable. The use of an external device, e.g. an oscilloscope that has to be manually connected and reconnected before measurements is not desired.

In the following, several different approaches are considered.

4.2.1. Comparison with software simulation

The most obvious way of establishing the hardware variations is to compare a hardware experiment to a software simulation. A possible setup would be to stimulate a neuron with a series of spikes sampled from a Poisson distribution and observe the dependence of the neuron's firing rate on the synaptic weight. The effective synaptic weight could be deduced from the corresponding software simulation. This method has been found to be impractical for several reasons. First of all, it relies on an optimal calibration of the hardware resting and reversal potentials and synaptic and membrane time constants. Measuring those separately would require additional effort and preventing an easy automation, because additional hardware such as an oscilloscope would have to be connected to the setup. Second, deviations of hardware behaviour from the mathematical neuron model would be completely neglected, thereby introducing systematical errors that are hard to account for.

4.2.2. Inverted Spike Threshold and Resting Potential

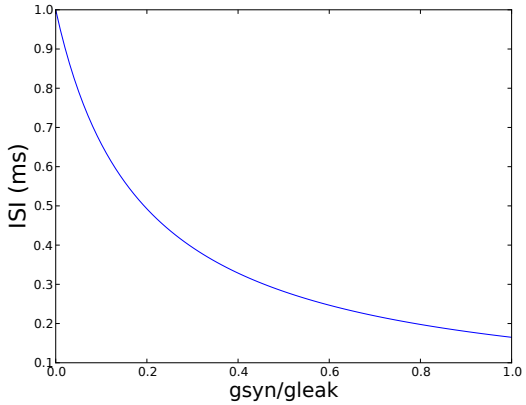


Figure 4.1.: Example of the dependence of the interspike interval of a periodically spiking neuron with $U_{\text{thresh}} < U_{\text{rest}}$ on a constant excitatory conductance g_{leak}, as described in section 4.2.2 and in equations 4.2 to 4.4. Used parameters: $\tau_{\text{refrac}} = 0$ ms, $\tau = 1$ ms, $U_{\text{reset}} = -70$ mV, $U_{\text{rest}} = -50$ mV, $U_{\text{thresh}} = -57.36$ mV, $E_{\text{rev,E}} = 0$ mV.

A second method is inspired by the neuron time constant calibration that is described in [*Brüderle*, 2009, Ch. 2.4.2]. In that case, the resting potential is set above the threshold, which leads to periodic spiking. The resulting interspike interval is given by equation 4.2.

(For simplification, the original method chose U_{thresh} appropriately, so the logarithm evaluated to 1.) The spike times can be used to deduce the membrane time constant with good precision without recording the membrane potential. As these two properties are desired, it was investigated whether the method could be adapted for a measurement of synaptic strength variations.

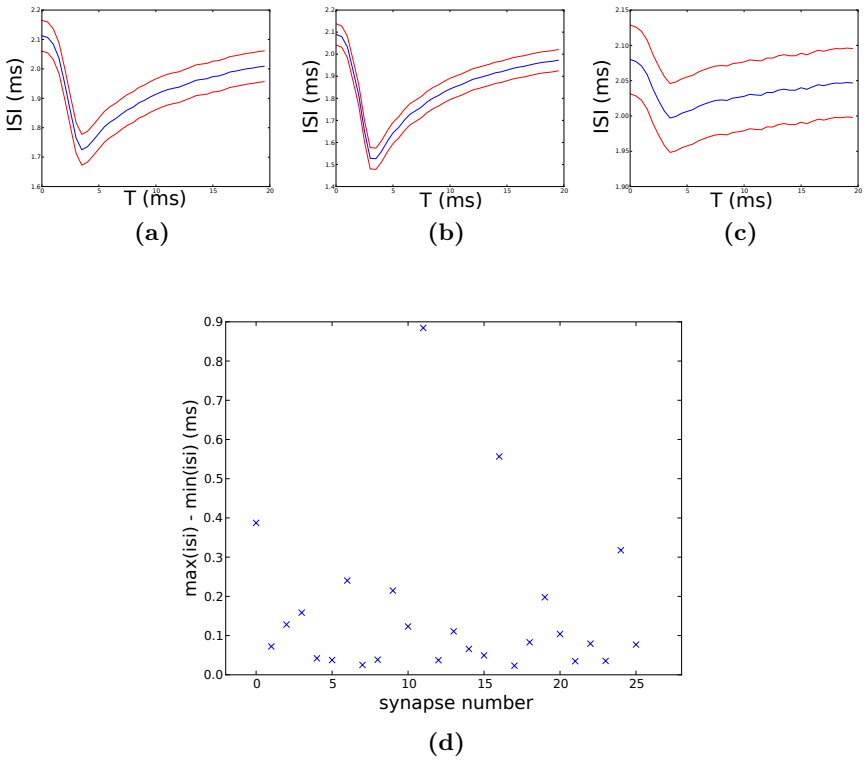


Figure 4.2.: (a) - (c) Examples of input-spike-triggered interspike intervals. 200 excitatory and 55 inhibitory background sources were connected to the randomly chosen neuron 44 of 192 available neurons on chip nr. 444. The background firing rate followed a Poisson distribution with a mean rate of 0.1 Hz for each connection. Each excitatory stimulus source was connected with a weight of .33 nS, each inhibitory source with a weight of 1 nS. The measured synapse fired with a constant rate of 10 Hz and was connected with a weight of 5 nS. The neuron was configured with $g_{\text{leak}} = 200$ nS, $U_{\text{reset}} = -75$ mV, $U_{\text{rest}} = -60$ mV, $U_{\text{thresh}} = -70$ mV. Each experiment ran for 48 seconds (biological time), meaning that each point is the average of 480 interspike intervals. The abscissa shows the time after the spike from the measured synapse, in milliseconds (biological time). Each graph shows the mean (blue line) interspike interval and the standard deviation (blue line), in milliseconds (biological time). (a) Synapse index 0, (b) synapse index 16, (c) synapse index 18 (d) Maximal change induced in the mean interspike interval by a synapse for the first 26 synapses.

Because synaptic input modifies the effective time constant and resting potential, as given by equations and for the case of a time-independent synaptic input, the variation of the interspike interval in the presence and absence of synaptic input from a single synapse will provide the needed information about the synaptic strength. As seen in figure 4.1, this method is most sensitive for small synaptic conductances. However, it has to be taken into account

that the total conductance follows the time course imposed by the synapse instead of being constant. Thus, the emulation has to run long enough to average out these fluctuations.

Finally, this method, also suffers from the fact that a translation between interspike interval and mean conductance, as shown in figure 4.1, is needed to get an actual result; this translation is dependent on the actual voltages and time constants which have to be measured independently or estimated, sacrificing precision.

The most important reason to consider a different method is that setting the spiking threshold below resting potenital is an atypical experiment setup that is not expected to be employed often, especially not in combination with synaptic input. For a more realistic configuration, a neuron should be stimulated by many different synapses, using a balanced combination of excitatory and inhibitory input.

$$\tau \dot{U} = -(U - U_{\text{rest}}) - \frac{g_{\text{syn}}}{g_{\text{leak}}}(U - E_{\text{rev,E}}) \tag{4.1}$$

$$T_{\text{isi}} = \tau \log\left(\frac{U_{\text{reset}} - U_{\text{rest}}}{U_{\text{thresh}} - U_{\text{rest}}}\right) + \tau_{\text{refrac}} \tag{4.2}$$

$$\tau \to \frac{\tau}{1 + \frac{g_{\text{syn}}}{g_{\text{leak}}}} \tag{4.3}$$

$$U_{\text{rest}} \to \frac{U_{\text{rest}} + \frac{g_{\text{syn}}}{g_{\text{leak}}} E_{\text{rev,E}}}{1 + \frac{g_{\text{syn}}}{g_{\text{leak}}}} \tag{4.4}$$

During deliberations about the advantages and disadvantages of the jitter method the question arose whether it could be adapted as a spike based measurement of the conductance time course. By stimulating a neuron using a single synapse with a constant rate, the neuron's firing rate is modified according to the synaptic conductance that applies at any given moment. Given a chosen set of parameters that causes the neuron to spike with a high firing rate, the synaptic time course is effectively sampled with the neuron's spiking rate. Using appropriate parameters, especially a high leakage conductance, a spiking rate of 500 Hz can be achieved, while the synaptic conductance time course has a temporal extent of more than 10 ms. Figure 4.2 (a) - (c) shows three examples of the synaptic time course averaged over 480 trials with an identical stimulus. Note how the synapse with index 16 (b) induces a strong decrease, while the synapse with index 18 (c) has a much smaller impact. On the other hand, the difference is mainly in the magnitude, not in the shape of the synaptic time course. The extent of this variation can be seen in Figure 4.2 (d). A few synapses with a very strong effect and many with a moderate or small effect are observed among the first 26 synapses. This picture gives a qualitative overview of the strength variation. Unfortunately, a quantitative comparison can not be made directly because a calibration curve as the one shown in Figure 4.1 depends on the precise knowledge of the configured resting, threshold and reversal potentials and the membrane time constant, which are not easily accessible.

4.2.3. Weight Dependence of Neuronal Firing Rates

The insights from the previous chapters are now combined to form a weight jitter measurment method that conforms to the desired criteria that have been established in the introduction of chapter 4.

First, the weight jitter is defined using the assumption that the only difference between synapses is the maximal weight that they can convey. This is supported by the conductance

course estimation that is described in the previous section. It is further assumed that the digital scaling of this weight by each synapse circuit is linear. With the previous methods, a quantitative statement was made difficult by the need for additional measurements, e.g. of the leakage conductance or resting and threshold potentials, and by the assumption that hardware neuron behaviour is ideal. Both of these issues are remedied by a setup in which the digital weight of a synapse is varied, changing the firing rate of the neuron it is connected to. This change is compared to a reference response curve, as illustrated in Figure 4.3.

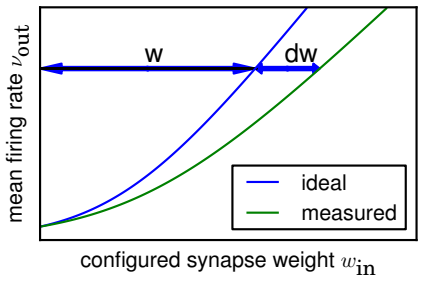
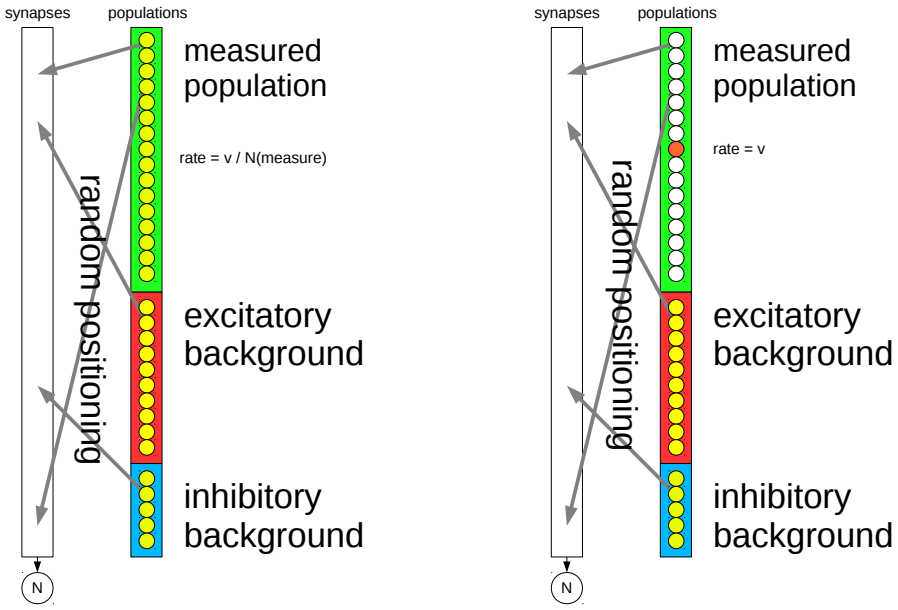


Figure 4.3.: Schematical illustration of jitter measurment.



(a) **(b)**

Figure 4.4.: Illustration of jitter measurement setup that is described in section 4.2.3. (a) Setup for the measurement of the reference curve, (b) Setup for the measurement of the strength of a single synapse. See text for further details

The exact implementation is illustrated in Figure 4.4. Three populations are chosen from the available synapses that connect to a neuron. For this experiment, an excitatory and an inhibitory background population were chosen that provide a level of activity that may be expected in a real experiment. These populations act as a load on the synapse drivers, raising the neuron's mean membrane potential. The number and configuration of synapses are given in table 4.1 (a) and (b).

The third population is the set of synapses whose strength will be measured. The type can either be excitatory or inhibitory. In each run, a Poisson spike train is sent over one synapse in this population, while the synapse's digital weight is set to different values. This produces the response curve for the synapse. To obtain the reference response curve, all synapses in the third population are connected to the neuron and each transports a Poisson spike train. The total firing rate of the spike trains is equal to the firing rate of the single synapse. To gain precision, more weight values are taken than in the single synapse case: Although each synapse can only take 16 distinct values, for a large population the values are rounded statistically to ensure the correct mean value. The resulting reference curve is fitted by equation 4.5 for excitatory and 4.6 for inhibitory synapses. The fit is necessary to determine the effective weight of each data point.

Each background synapse fires according to the Poisson statistics using a given synaptic weight and mean rate. These parameters are chosen such that the impact of the measured synapse from the third population is as large as possible. In case of the excitatory configuration, the values were chosen such that the neuron's firing rate is near zero with background only and much higher with the measured synapse set to its maximal conductance value. (See 4.1 for exact values.)

The synapses from the three populations are not mapped directly to the hardware, but shuffled randomly, in the same way for each run. This is done to avoid systematic errors due to possible regular deviations on the chip.

$$f_{\mathrm{exc}}(x) = \log\left(\exp\left(\frac{x-m}{w}\right)+1\right)\cdot w \cdot k \tag{4.5}$$

$$f_{\mathrm{inh}}(x) = \exp\left(\frac{-x}{a}\right)\cdot k + x\cdot s + b \tag{4.6}$$

The measurment results on a Spikey chip No. 444 (Version 4) are shown in Figure 4.5. Figures 4.5 (a) and (b) show that the fit functions in the excitatory and inhibitory case reflect the spiking behaviour in a satisfactory manner – the difference between fit and measured value lies well within the measurment error. Figures (c) and (d) show that the variation of synaptic strengths is indeed very strong; some synapses are stronger in the 1.5 nS setting than others at 5 nS in the excitatory case.

For each data point, the effective weight was calculated by taking the inverse of equation 4.5 resp. 4.6. Effective weights larger than the maximal possible synaptic weight are ignored, because in this region the functions are extrapolated. An average is taken, weighting each data point by the inverse of its error squared. This value is the effective weight of the synapse. The distribution of effective weights is shown in Figure 4.5 (e) and (f). The mean effective weight was 0.91 for the inhibitory and 0.71 for the excitatory case, with the standard deviation being 0.38 and 0.47, respectively. In the excitatory case, many synapses had an effective weight close to zero, because of the chosen background configuration. (A different configuration

which covered both, the strongest and weakest synapses could not be found in reasonable time.)

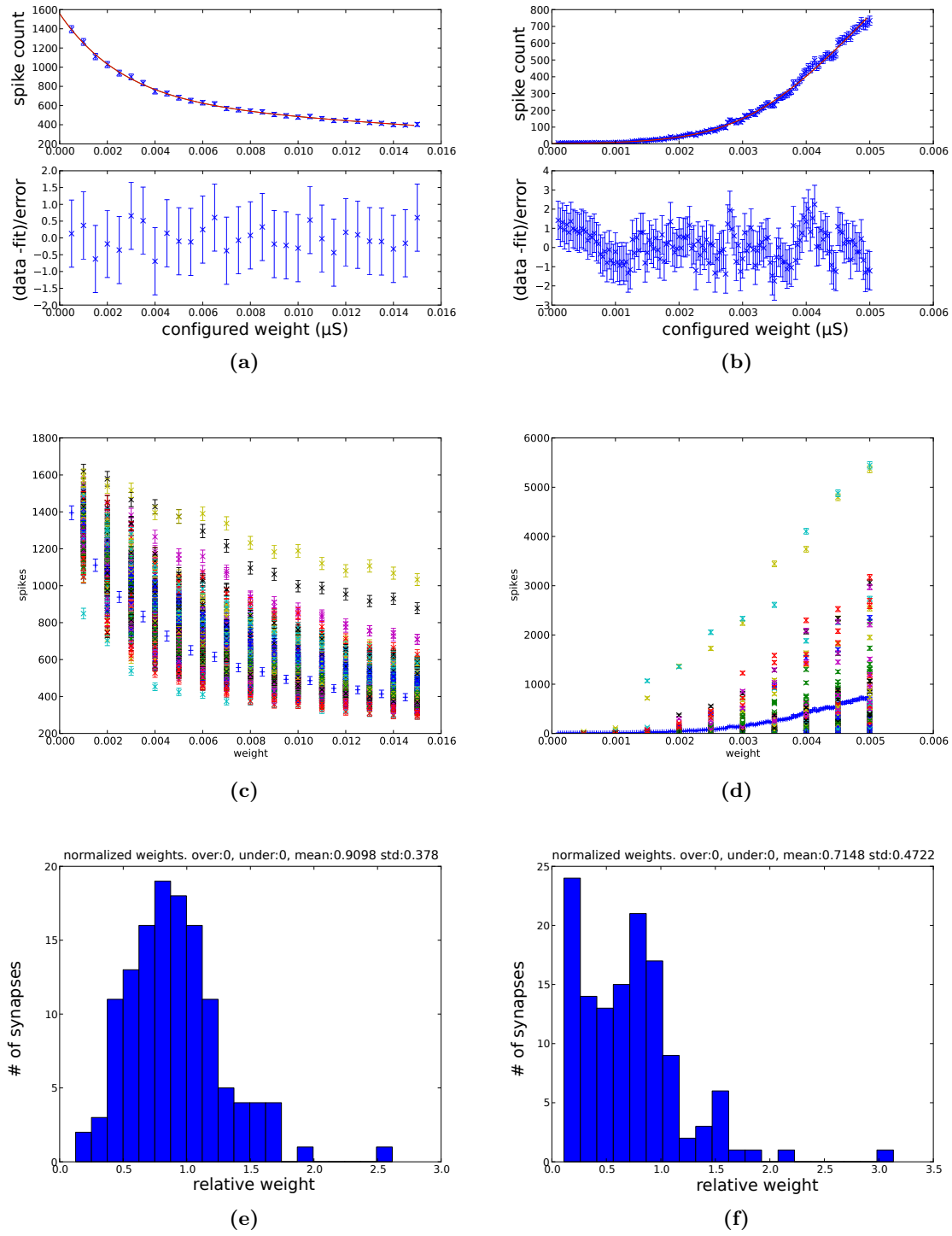


Figure 4.5.: Weight jitter measurement using neural response as indicator. (a), (b) Inhibitory and excitatory reference curves. Top: obtained data with fit. Bottom: difference of data and fit divided by measurement error. (c), (d) Stimulation by background and a single synapse, for 128 synapses. (c) Inhibitory setup. (d) Excitatory setup. (e), (f) Histogram of effective weights for inhibitory and excitatory setup. See text for further details.

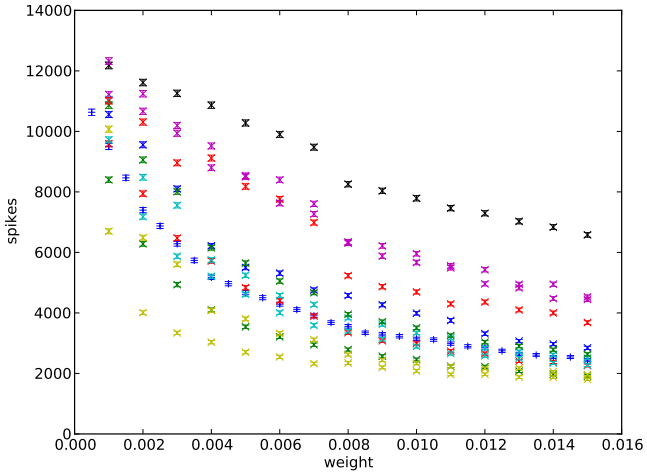


Figure 4.6.: Weight jitter measurement using neural response as indicator. Stimulation by background and a single synapse, for 128 synapses. Analogous to Figure 4.5 with increased experiment duration.

Several issues are noteworthy. First of all, a few outliers with an effective weight of 2 or more occur. These are the same for the excitatory and inhibitory measurement. Second, the average in both cases is smaller than 1. This means that a spike train that is distributed between all synapses has a stronger effect than the average of all synapses relaying the spike train individually. This is likely due to some very strong outliers in the relative weight distribution, and a frequency dependence of the synaptic efficacy. The second point is reasonable to assume because the conductance course that can be sent over one synapse is reset at the beginning of each spike while conductances from two different synapses can be added.

This discrepancy is not crucial to the main result because the setup that employs all synapses is mainly used to provide a good data set for the fitting routines. If a higher precision is required in future, a different reference may be considered, for example the average of the single-synapse runs. In the given case, the largest weight deviations stem from absolute strength differences between synapses; these differences are much stronger than effects that would arise from the aforementioned discrepancy. This is evident from Figure 4.5 (c) and (d).

The third issue is the non-linearity of single synapse circuits. Figure 4.6 has been generated analogously to 4.5 (c) with an increased simulation time and fewer measured synapses. This shows the influence of digital-to-analog converter (DAC) precision on the neural response. Because the DACs are implemented in a very space-saving manner, they introduce imprecisions which can, for instance, be seen in Figure 4.6 for the top-most black and red data set. In the first case there is a noticeable gap between the 7th and 8th data point (0.007 to 0.008 µS), in the second one the response magnitude jumps after every four increments. The cause for this is that the most significant resp. the second most significant bit of the DAC is stronger than expected.

It has to be noted that the given experiments were conducted on an uncalibrated Spikey 4 system. This means that the translation between software and hardware synaptic weights was done without regard for deviations of synaptic efficacies. The reason was that no calibrated Spikey 4 systems were available at the time of the writing of the thesis. This does not affect

the feasibility of the presented methods for the measurement of synaptic weight jitter. On the contrary, they can be used to cross-check a different calibration routine.

	exc. background	inh. backround	measured
mean rate (bio. Hz)	2	0.8	120
weight (μS)	.004	.001	-
synapse number	100	28	128

(a)

	exc. background	inh. backround	measured
mean rate (bio. Hz)	1.1	0.8	120
weight (μS)	0.002	0.002	-
synapse number	100	28	128

(b)

pyNN.setup parameters		neuron parameters	
`calibOutputPins`	True	`e_rev_I`	-80.0
`calibSynDrivers`	True	`tau_syn_E`	30.0
`calibTauMem`	False	`tau_syn_I`	30.0
`calibVthresh`	varying	`g_leak`	20.0
`loglevel`	2	`v_reset`	-75.0
`mapping_offset`	0	`v_rest`	-75.0
`rng_seeds`	[10298]	`v_thresh`	-65.0
`timestep`	0.1		

(c)

Table 4.1.: Parameters for final measurment using the method described in section 4.2.3. (a), (b) Mean rate and projection weight. The mean rate is given per synapse for background populations i.e. the total firing rate is the product of mean rate and synapse number. Time consumption was 60 minutes (real time) in (a) and 30 minutes in (b) (a) Excitatory setup, (b) inhibitory setup, (c) global and neuron hardware configuration. Neuron parameters are given in the units defined by *PyNN*.

4.3. Conclusions and Outlook

The need for an estimate of weight jitter on a neuromorphic hardware system led to the investigation of several possible measurement methods. The principal idea is that a method that compares the effects of synaptic jitter to a reference that stems from the hardware itself is superior to a comparison between hardware and software simulations, because fewer assumptions about the operation of the hardware and the precision of calibration need to be made. Using such a method, the jitter value for an uncalibrated Spikey was estimated to be 40% with the main source being the synaptic drivers. The digital-to-analog converters in each synapse circuit have a weaker, but measurable effect. The value of 40% is taken as the upper bound for the software simulations in the following chapters. The HICANN module that will be used on the wafer-based hardware system employs a different synapse architecture that is expected to exhibit less variability. (See section 1.2.1)

As the presented methods are purely based on spike trains rather than membrane potential recordings, they can be considered as good candidates for calibration purposes, for the same reasons as the ones mentioned in the beginning of the chapter. Comparing the final presented measurement method to the given criteria, one finds that the execution *speed* was not very high – at a duration of 30 to 60 minutes for 128 synapses, the method would extrapolate to several weeks for a single Spikey chip. However, due to the fact that the method is purely rate-based, which, in computational terms, amounts merely to counting spikes, all the computations performed offline, after the experiment, should be easily implemented into FPGA logic if a variation of the presented method is considered for use with the wafer-scale system. This would reduce the computation time by several orders of magnitude.

The *expressiveness*, i.e. a clear relation to the weight jitter definition given in 3.2.2 was perfect, as the exact value was measured. The *precision* was sufficient for the given case due to a large deviation of effective synaptic weights, so measurement errors for single synapses were much smaller than the overall distribution width. The setup was *realistic* in the sense that background stimulation was applied. However, the total amount of stimulation can not be chosen too strong, because otherwise the firing rate change induced by the measured synapse can not be detected. Because the method is purely spike-based, it can be *automated* more easily than a method relying on recording of membrane potentials.

If the method of comparison with a hardware-generated reference is adapted for calibration purposes, further optimizations concerning the choice of the working point and the execution speed have to be performed. The working point, i.e. the exact choice of background stimulus and neuron parameters, was chosen by trial-and-error in the presented investigation. A rigorous consideration may provide useful insights as to which configurations yield the best results in terms of effective weight resolution and realism in relation to standard use cases. Furthermore, the experiment duration will have to be minimized, for example by using only a few synaptic weight values for the actual calibration and a complete sweep as a cross-check.

Parallelization of the method by simultaneous measurement of several neurons at once also represents a possible optimization.

5. Analysis of Benchmark Models

In this chapter, two Demonstrator benchmark models are investigated with respect to their behaviour in the presence of hardware distortions. The analysis of the *synfire chain* model focuses for a large part on the necessity of synaptic delays and possible workarounds to replace missing delays by parameter modifications. The analysis of the *Self-Sustained Asynchronous Irregular States*-models shows the effects of hardware distortions on random networks of neurons with complex behavior.

5.1. Synfire Chain with Feed-Forward Inhibition

In this chapter, the behavior of the synfire chain model in the presence of hardware distortions is investigated. It is shown to what extent the original functionality can be regained by changes of the model parameters. While most hardware-specific effects either have little influence or are easily compensated, a possible absence of delays emerges as having the deepest impact on the model functionality.

5.1.1. Motivation

One of the key questions of neural research is the nature of information processing within neuronal circuits. An essential component to this question is the understanding of the *neural code*, the way in which data is represented and processed by neuronal activity in a given network. The key concepts that must be explained by a given neural code are, as defined by Perkel and Bullock (*Kumar et al.* [2010]), *stimulus representation*, *interpretation*, *transformation* and *transmission*.

Stimulus representation is the relation between an external stimulus and neural activity. Interpretation is the way in which a processing circuit can retrieve information from the stimulus representation. Transformation encompasses the mechanism by which a network performs a computation on the input and produces the result in the given code. Transmission is the passing of information from one region of the brain to another. Efficient information transport is an essential part of neural computation and has been analyzed by simulations and theoretical investigations (*Diesmann et al.* [1999], *Goedeke and Diesmann* [2008]). One possibility to implement a transport mechanism is a feed-forward network with a convergent-divergent connectivity: neurons in each layer receive many synaptic inputs from the previous layer and project to many neurons in the next layer.

Such a network was employed by *Abeles* [1991] under the name "synfire chain" as an explanation of precise spike timing that was observed in awake animals.

In connection with the neural coding problem (*Dayan and Abbott* [2001]), e.g. whether information is encoded by precise spike timing or merely by a neuron's firing rate, the conditions under which a given code can be *transported* offer a valuable information. *Kumar et al.* [2010] shows that asynchronous firing rates and synchronous pulse packets can be transported by a

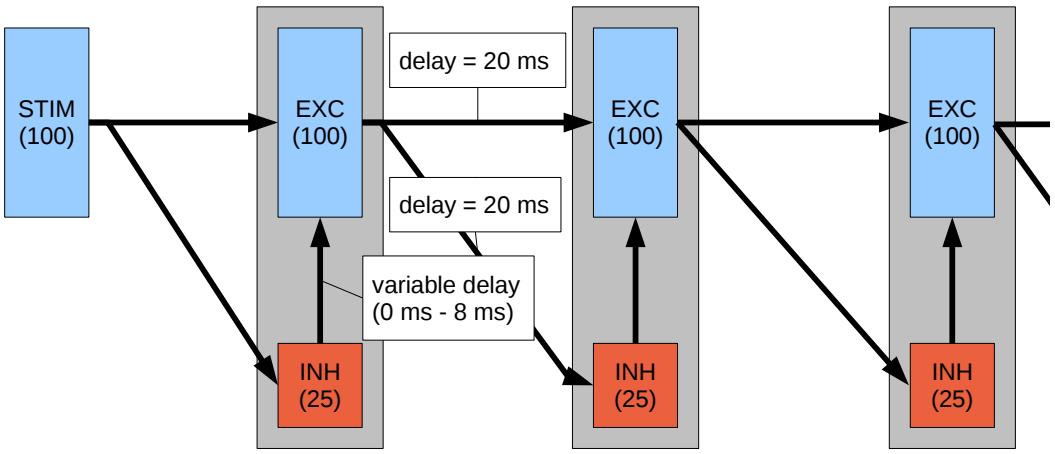


Figure 5.1.: Schematic of the synfire chain model. Detailed description in text.

synfire chain depending on the connection density between consecutive groups. For certain values of these parameters, none, only one or both modes of propagation are possible.

The synfire chain benchmark model that is employed within the FACETS Demonstrator addresses the question, to what extent feed-forward inhibition affects the filtering properties of signal propagation in a synfire chain. In the case of feed-forward inhibition, an inhibitory neuron projects onto an excitatory neuron; both are driven by a third excitatory neuron. The abundance of this type of connectivity in the central nervous system, the observation that inhibition and excitation are correlated in the cortex (*Okun and Lampl* [2008]) together with the fact that inhibitory connections are usually local pose a strong motivation to investigate the filtering properties of a synfire chain that incorporates a feed-forward inhibition scheme.

Beyond the scope of pure signal transportation, the synfire chain concept can be used to model computation paradigms like logic gating (*Vogels and Abbott* [2005]) and compositionality, i.e. the hierarchical representation of parts (*Abeles et al.* [2004]).

5.1.2. Network model definition

The model defines a number of neuron groups that are arranged in a sequence, as can be seen in Figure 5.1. Each group consists of a population of 100 excitatory and a second population of 25 inhibitory LIF-neurons. The inhibitory population projects to the excitatory population of the same group while the excitatory population projects to both populations of the subsequent group. Each neuron receives a fixed number of incoming synapses from both populations. The connection count and synaptic weights are listed in table 5.1 (b). To simplify the evaluation, the background noise for the model was chosen to induce a membrane fluctuation without inducing background activity. This was done in analogy to the noise for the *FFI circuit* in the original publication in *Kremkow et al.* [2010].

Concrete Implementation of Distortions

The modeling of distortions, as described in section 3.2, is applied to the synfire chain model. The types of considered distortions are reiterated shortly:

C_m	0.29 nF
$E_{\text{rev,E}}$	0 mV
$E_{\text{rev,I}}$	-75 mV
τ_m	10 ms
τ_{refrac}	2 ms
$\tau_{\text{syn,E}}$	1.5 ms
$\tau_{\text{syn,I}}$	10 ms
V_{reset}	-70 mV
V_{rest}	-70 mV
V_{thresh}	-57 mV
V_{init}	-70.0 mV

(a)

projection	EXC → EXC	EXC → INH	INH → EXC
synaptic weight	1 nS	3.5 nS	2 nS
default delay	20 ms	20 ms	0 ms - 8 ms
incoming synapses per neuron	60	25	25

(b)

Table 5.1.: (a) Neuron parameters used for excitatory and inhibitory neurons. (b) Connectivity within the model.

synapse loss A synapse loss value of p means that each synaptic connection has a probability p of not being realized.

weight jitter A weight jitter value of j means that each synaptic weight with original value w is sampled from a Gaussian distribution with mean w and standard deviation $j \cdot w$

synaptic delays The delays of the model can, in principle, be set for each type of projection individually. Because of the feed-forward structure of the network, the signal propagation properties are only affected by the delay difference between the two intergroup projections, not by the absolute value. Thus, in most cases, variation of the *local delay* between the populations of one group was sufficient.

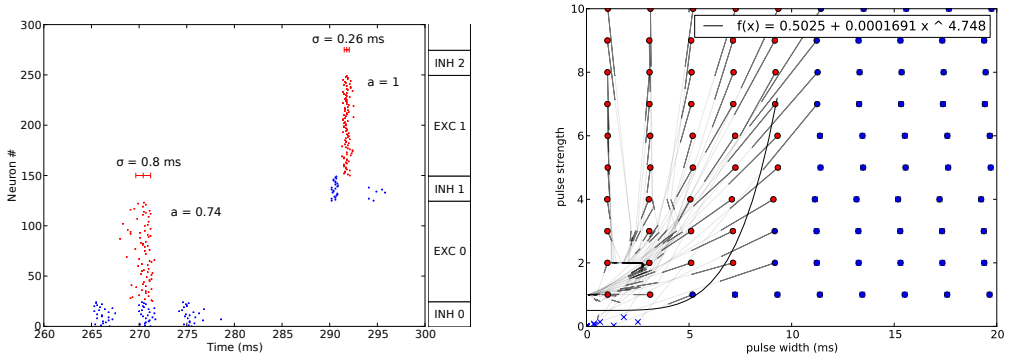
These parameters were incorporated in the model by implementing a custom *PyNN* `Connector` as a replacement for the built-in `FixedNumberPreConnector`. This new connector takes an additional parameter, the synapse loss probability, and randomly establishes connections with a probability of $1 - p$. The advantage of this approach is that it makes use of the pyNN abstraction to prevent unnecessary repetitions.

5.1.3. Definition of Functionality Characteristics

The quantities that are used to characterize distortion-induced changes need to be carefully defined.

Initially, the occurrence and absence of activity propagation was considered and rejected as a functionality characteristic, because it would neglect the effects on the filtering properties of the network.

These properties can be described by measuring the response strength and temporal spread of the synfire pulse in each group. In the following, the strength a denotes the mean number of spikes per neuron in the excitatory population and the temporal spread σ denotes the standard deviation of all excitatory spike times of a synfire pulse (See Figure 5.2 (a)). It is common to represent the propagation of the pulse as a trajectory in the (σ, a) state space (*Kumar et al.* [2010]).



(a) **(b)**

Figure 5.2.: (σ, a) state space of the synfire chain model. (a) Raster plot of the synfire chain activity. The first group was stimulated by a Gaussian pulse packet with a temporal spread of $\sigma = 4$ and one spike per neuron ($a = 1$). The response of the first two groups is shown. The inhibitory population responds earlier due to the stronger synaptic weight $g_{\mathrm{EXC}\to\mathrm{INH}}$. The rapid contraction of the pulse packet width is caused by strong local inhibition that stops excitatory activity. The inhibition takes effect after a delay, which amounts to 6 ms in this case. (b) Visualization of the filtering properties of the synfire chain. The filled circles represent the (σ, a) parameters of the stimulus. Red circles represent stimuli that lead to a detectable activity in the excitatory population of the sixth synfire group; otherwise the circles are coloured in blue. The trajectory in state space is encoded by grey lines; The start of each line is marked by a black line for better visibility.

The initial point denotes the (σ, a) parameters of the stimulus pulse; each consecutive point, shown as an arrow, denotes the activation of one synfire group. Important properties that become apparent in this visualization are the location of fixed points (one near $\sigma = 0, a = 1$, in which case the propagation continues indefinitely, and one at $a = 0$ i.e. the propagation dies out) the location of the separatrix between those stimulus conditions that lead to a stable propagation and those that evoke only a short or no response. Initial conditions that lead to a stable propagation are shown as red circles; a stable propagation is assumed when activity is detected in the excitatory population of the last (the sixth) group. In analogy to the original publication, the separatrix is fitted by a function $f(\sigma) = a + b \cdot \sigma^c$ using the rightmost points that lead to a stable propagation. The range of σ and a was also adapted from *Kremkow et al.* [2010].

5.1.4. Replacing Random Current by Synaptic Background

The model definition (5.1.2) employs a Gaussian background current to introduce trial-by-trial variability. The current emulates the background spiking activity that is seen by neurons *in vivo* (*Destexhe et al.* [2003]). The wafer-based hardware system is not equipped with a sufficient number of background current sources to provide independent stimulation for a large number of neurons. On the other hand, each HICANN module offers eight efficient pseudo-random poisson spike source generators. Therefore, synfire chain model is modified

to use a spiking background. This modification is taken as an opportunity to study the limits that are imposed on such a replacement.

The following conditions must be satisfied by the background stimulus:

- The *model definition* requires that the background stimulus raises the mean membrane potential without inducing background spiking. Equivalently, the mean potential must lie several standard deviations below the spiking threshold.
- The *hardware implementation* limits the maximal firing rate that is sent from one HICANN to 12.5 kHz at a speedup factor of 10^4. This includes the spikes from 56 neurons and eight poisson spike sources that are located on the HICANN module.

The second condition leads to the choice of a background rate of 2 kHz per neuron, leaving 10.5 kHz for interneuron communication. Figure 5.4 shows that the first condition is fulfilled by a synaptic weight of 1 nS. As can be seen in 5.4 (b), these values produce a near-zero firing rate while raising the mean membrane potential to -64 mV. In comparison, a stimulus with a 10 kHz firing rate and a synaptic weight of 0.3 nS raises the mean potential to -60 mV, the same value as for the originally injected background current.

Correlated Background

Because the number of neurons is larger than that of the background sources, a fully independent stimulation is not possible. Sources need to be shared between the neurons in a way that minimizes the stimulus correlation. Furthermore, only the sources from a few HICANNs should be used for each neuron to prevent an additional routing overhead. Given a configuration with 64 neurons per HICANN and a feedforward network with 100 excitatory and 25 inhibitory neurons per group, an efficient placement solution would map each group onto three HICANNs (two for the excitatory and one for the inhibitory population of the group), and use the L1 bus connections that are already configured to transport spikes from one group to its sucessor to also transport the background stimulation. Given the current placing limitation that neurons and sources on one HICANN must be either all excitatory or inhibitory this leads to a suggested use of 16 background sources for 125 neurons.

To find a distribution scheme that minimizes the background correlation, a custom, semi-heuristic algorithm is employed (*"Avoiding correlations in neural activity on neuromorphic hardware"*, Mihai A. Petrovici et. al., to be published). Table 5.2 shows how many subsets of sources have been found by the algorithm with the given restrictions. k is the size of each subset and m is the maximal number of sources that can be shared by two subsets. The cross-correlation of two membrane potentials is expected to be equal to the number of shared background sources divided by the total number of sources. The underlying assumption for this estimate is that the time course of the membrane potential is well approximated as a sum of postsynaptic potentials. As can be seen in Figure 5.3a, the assumption is reasonable: The peaks of the correlation coefficient at 0, 1/4 and 2/4 are the result of membrane potential pairs that share zero, one or two of the four background stimuli, respectively. The final choice for the distribution scheme is represented by $N = 16, k = 4, m = 2$ since it offers the best ratio $\frac{k-m}{k}$ of all possibilities with at least 125 found input pairs, and is therefore expected to minimise background correlations.

A more important question is the influence of the background correlation on the signal propagation in the synfire chain. Correlated background is expected to increase the synchrony

N	k	m	found (best)	
16	2	1	120	
16	3	1	35	
16	3	2	**560**	
16	4	2	**127**	←
16	4	3	**1820**	
16	5	2	37	
16	5	3	**268**	
16	6	3	53	
16	6	4	**481**	
16	7	4	65	
16	7	5	**667**	
16	8	6	**756**	

Table 5.2.: Input distribution found by the custom, semi-heuristic algorithm. The number of found input configurations with k inputs, of which each pair shares at most m inputs. Configurations with sufficient (≥ 125) configurations are shown in bold. The selected configuration is marked by an arrow (←)

of the group activity. Figure 5.3a shows that while identical background stimulus leads to a completely synchronous activity, because each neuron receives exactly the same input, strongly and moderately correlated background stimulus introduce a temporal spread that is of the same order of magnitude as for completely uncorrelated case. Because the simulation time step was 0.1 ms, which is close to the pulse width, a more quantitative statement can not be made.

5.1.5. Weight Jitter

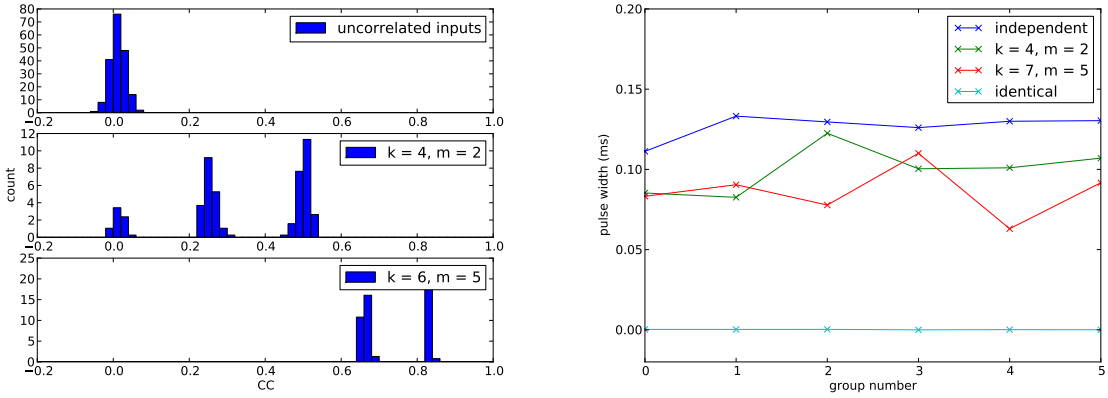
The influence of weight jitter on the model is minimal at values of less than 50%. (The investigation of the chip-based neuromorphic system provided an estimation of maximal weight jitter of 40%, as described in) Figure 5.5 (a) shows the response of the synfire chain to a stimulus with $a = 2, \sigma = 3$ ms. The introduction of weight jitter of up to 30% does not significantly affect the width of the excitatory spike response. Even a very large weight deviation of 80% of the original value does not introduce a large variability. The response in the state space also does not show significant differences; the stable region stays essentially the same at 30% and 80% weight jitter, while the region in which at least a response in the first group is noted, only grows notably for a value of 80%.

Due to this minimal effect, no compensation for synaptic weight jitter is considered.

5.1.6. Synapse Loss

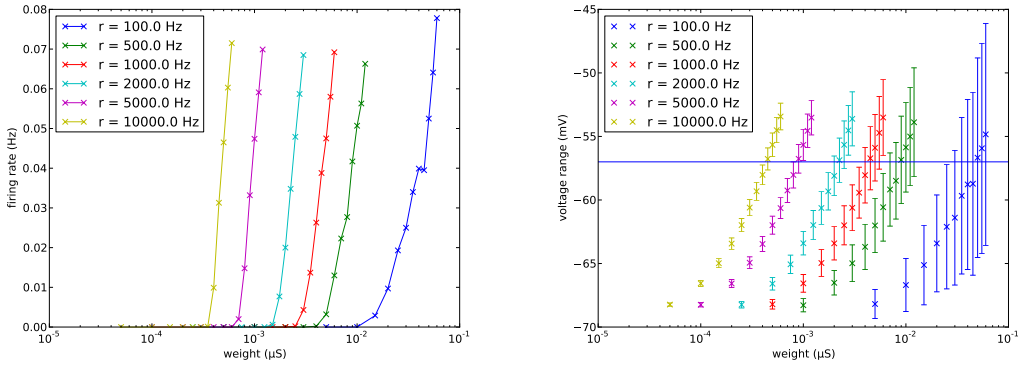
Effect of Synapse Loss

To quantify the effects of synapses being lost in the mapping process, the response of the synfire chain to a stimulus with $a = 4$, $\sigma = 2$ ms was tested at different values of synapse loss. (Figure 5.6) Up to a loss of 30%, the propagation can be sustained for at least six groups. At 40%, the activity can no longer be sustained. The width of the pulse packet increases with synapse loss, because a reduced connection density leads to a weaker correlation between the



(a) **(b)**

Figure 5.3.: (a) Histogram of correlation coefficients using different distribution schemes. (b) Temporal pulse width of the synfire pulse packet depending on the level of background correlation.



(a) **(b)**

Figure 5.4.: Effect of Poisson background stimulation on a LIF neuron with parameters equal to those in 5.1. (a) Firing rate of the neuron as a function of synaptic weight for different mean firing rates of the stimulus. (b) Mean and standard deviation of the membrane potential with disabled spike mechanism. The horizontal line denotes the original spiking threshold.

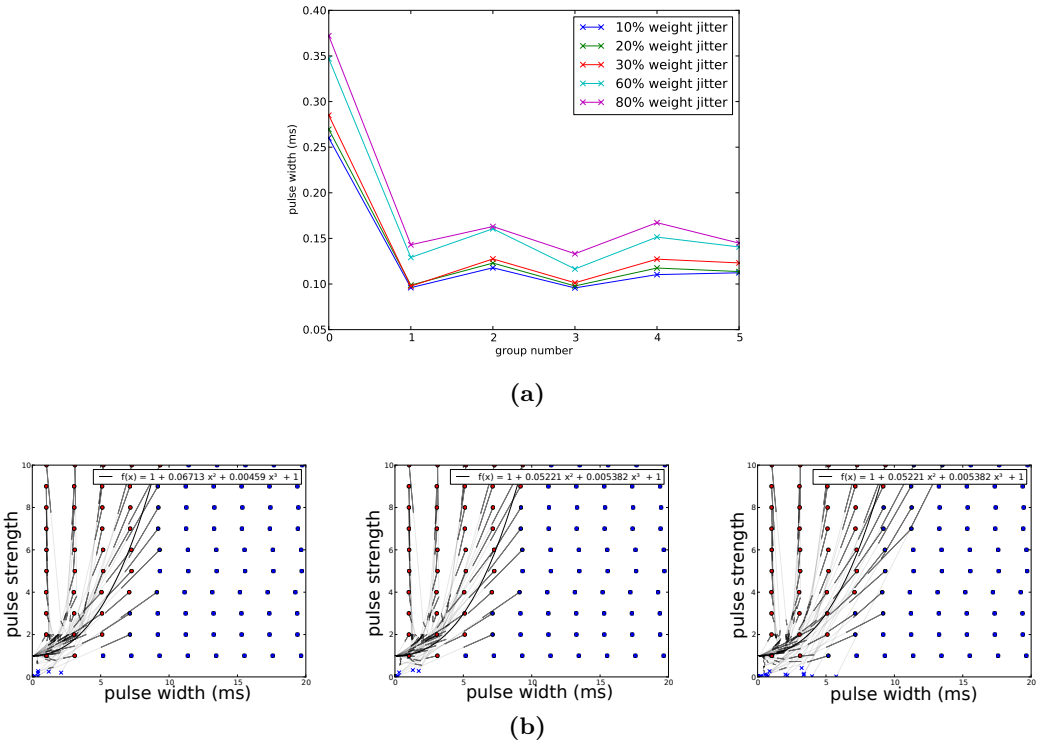


Figure 5.5.: Influence of weight jitter on the synfire chain state space. (a) Pulse packet width σ in a synfire chain that was stimulated by a pulse with $\sigma = 3$ ms and $a = 2$, under different values of synaptic weight jitter. (b) State space for 10%, 30% and 80% of weight jitter (from left to right).

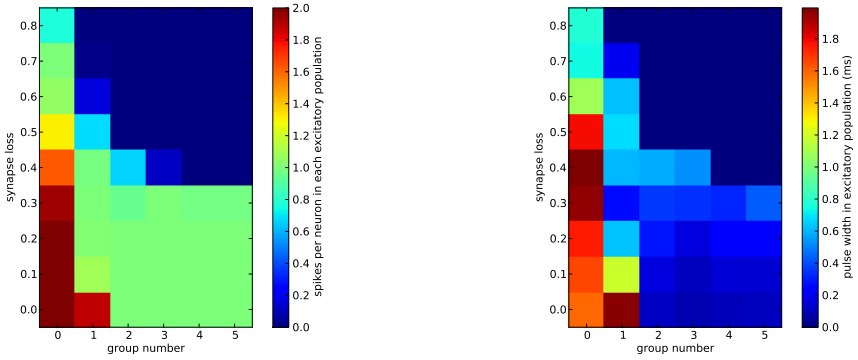
stimulus that is seen by the individual neurons; additionally, the total conductance injected into a neuron is smaller which leads to a slower response and thus allows a stronger influence of the stimulus and background variability on the exact spike time of each neuron.

Compensation Mechanism and Value Constraints

The most obvious idea to prevent the extinction of the propagation is to increase the connection weight keeping the product of weight and mean number of remaining connections constant:

$$w_{\text{compensated}} = w_{\text{original}} \cdot \frac{1}{1 - p_{\text{loss}}} \tag{5.1}$$

For a completely synchronous stimulus and a large population size this provides a perfect compensation, because of the linearity of the conductance term. For smaller group sizes, the variability of the connection count, given by the Poisson-distribution, gains influence. For broad stimulation pulses, the connection density plays an important role: A feed-forward network consisting of excitatory neurons enables the propagation of a firing-rate code *Kumar et al.* [2010] for sparse and strong connectivity, and the propagation of synchrony for dense and weak connectivity. First, the validity of this compensation rule has to be tested. For this purpose, the initial experiment was repeated while scaling all connection weights by the same factor and introducing synapse loss as described in 5.1.2.



(a) **(b)**

Figure 5.6.: Response of the synfire chain model to a stimulus with $a = 4$, $\sigma = 2$ for different values of synapse loss. Stable propagation ceases at 40% lost synapses. (a) Mean number of spikes in each excitatory population. (b) Pulse width in each excitatory population. The pulse width is set to 0 in cases where no spikes occurred.

Figure 5.7 shows the characteristics of the activation in the first and last groups for the excitatory populations together with the ideal compensation (green curve). The spiking activity in the first (a) and last (b) group can be restored for synapse loss values of at least 70%. The fact that the ideal scaling rule seems to be too weak, i.e. the ideal curve in 5.7 (b) approaches the lower bound of the area with exactly one spike per neuron. This is attributed to the fact that each neuron receives only 60 excitatory synapses without synapse loss and this number is small enough that the assumption of a large population size does not hold. Nevertheless, the scaling rule is taken to be adequate for small and moderate synapse loss values.

Finally, the compensation rule that has been defined above is tested in the default stimulus region of $a \in [1, 10]$ and $\sigma \in [0\text{ ms}, 20\text{ ms}]$. For this test, the focus is kept on two aspects: First, the region in the a, σ state space in which excitatory activity can be detected in all groups, which will be called *s*table region. Second, the region in which such activity can be detected in at least the first group, which will be called *r*esponse region. In the case of a weak stimulation, increasing synapse loss causes both regions to shift towards larger values of a (Figure 5.8 (c)). The stable region disappears at a critical value of synapse loss at which its border passes the stable fixed point. In the case of strong stimulation (Figure 5.8 (c)), synapse loss of 30% does not significantly affect the stable region, while the response region protrudes to larger σ. The effect at small a is attributed to the decreasing excitatory stimulus of the excitatory population of each group, while the second effect is a result of decreasing filtering due to diminishing local inhibition.

The effect of the compensation is shown in Figure 5.8 (b) and (d). Notably, the stable region can be qualitatively restored to the undisturbed case for weak and strong stimulation even for 60% synapse loss. However, the response region is notably larger in this case. The convergence towards the fixed point is also slower, as can be seen in the spread of the location of the second synfire group in the state space in Figure 5.8 (d). The reason is the sparsity and inhomogeneity of the connectivity at high synapse loss.

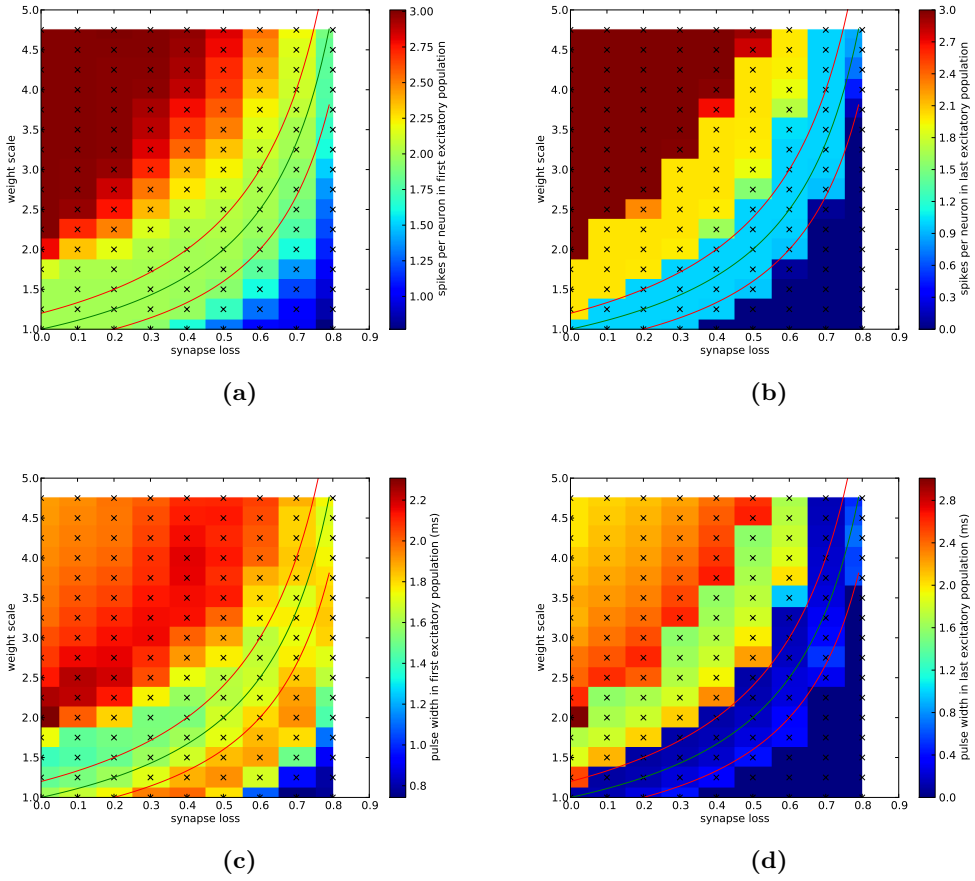


Figure 5.7.: Response strength and width of the first ((a), (c)) and sixth ((b), (d)) synfire chain group. The green line denotes the "ideal" compensation as defined in the text, the red lines denote deviations from the "ideal" compensation by ±20%.

5.1.7. Synaptic Delays

A distinctive feature of the synfire chain model and one decisive factor for its inclusion in the FACETS Demonstrator benchmark library is its dependence on synaptic delays. As the possibilities for realizing synaptic delays on the wafer-scale hardware are limited to connections that include *Layer 2* communication, it is essential to analyze whether the original model behavior can be restored in the case of absent delays by changing other parameters.

Effect of Local Delay on Synfire Propagation Properties

Figure 5.9 shows the influence of the magnitude of local delay on the location of the stable region. With minimal delay, no stimulus can evoke a stable propagation, because the excitation is prevented by the early onset of local inhibition within a group. With increasing delay, the stimulus width σ that can still evoke a stable propagation also increases. For the minimal delay value, no stable propagation occurs (Figure 5.9 (a)).

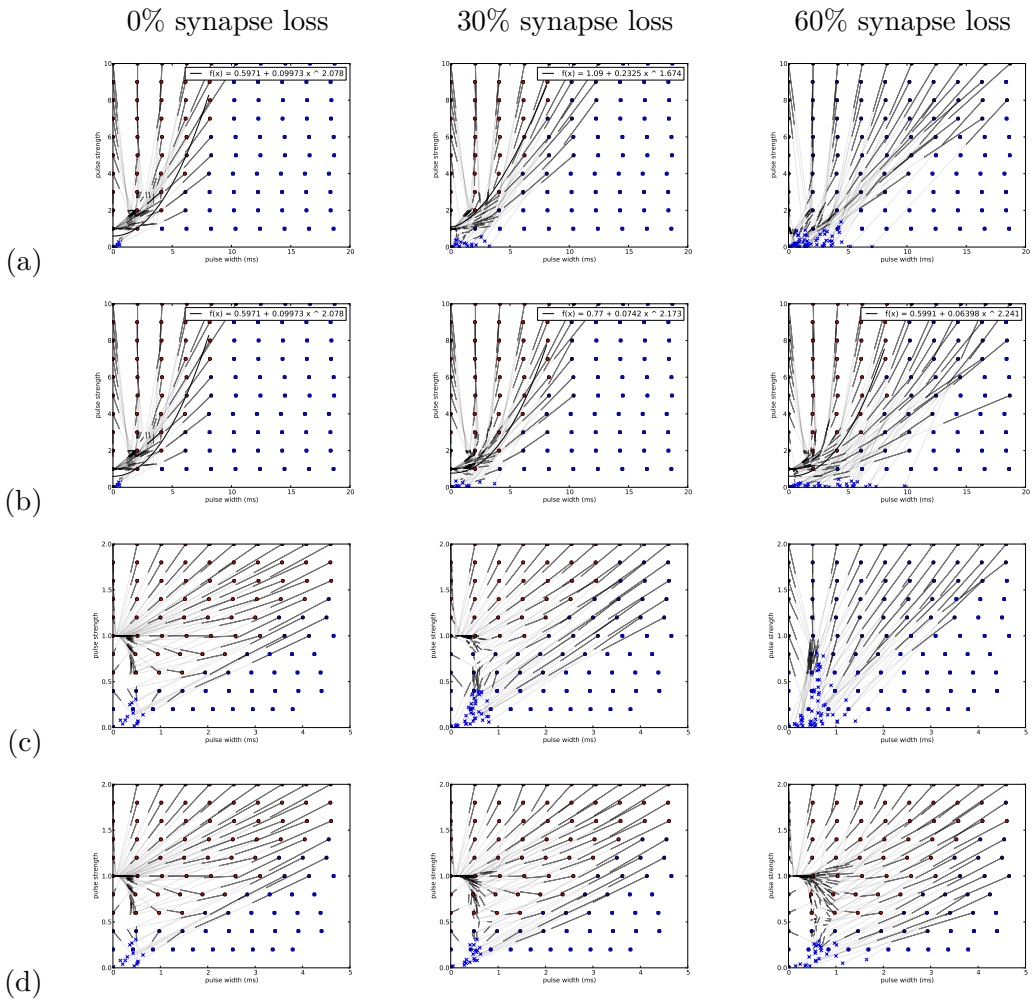


Figure 5.8.: State space behavior of the synfire chain model for different values of synapse loss. (a) Synapse loss applied to original model. (b) Synapse loss compensated by weight scaling (See section 5.1.6 for details) (c), (d) Zoomed versions of (a) and (b) depicting the convergence behaviour of trajectories close to the two fixed points.

Strategy for the Reintroduction of Delays

There are two goals that are addressed in this section. The first is to show that changes of synaptic and neuronal properties can cause a shift in the separatrix in the (σ, a) space similarly to the effect of changing synaptic delays. The second aim is to quantify the amount of delay that can be achieved by such compensation methods.

Before compensation methods can be investigated in detail, the large set of possibilities has to be limited to allow a systematic investigation. The following considerations provide a basis for the chosen compensation mechanisms:

- Small changes to the model are preferred. An arbitrarily complex change can evoke arbitrarily complex behavior, while diminishing the generalizability of the experiment. Major conceptual modifications such as the introduction of additional interneurons were therefore discarded.

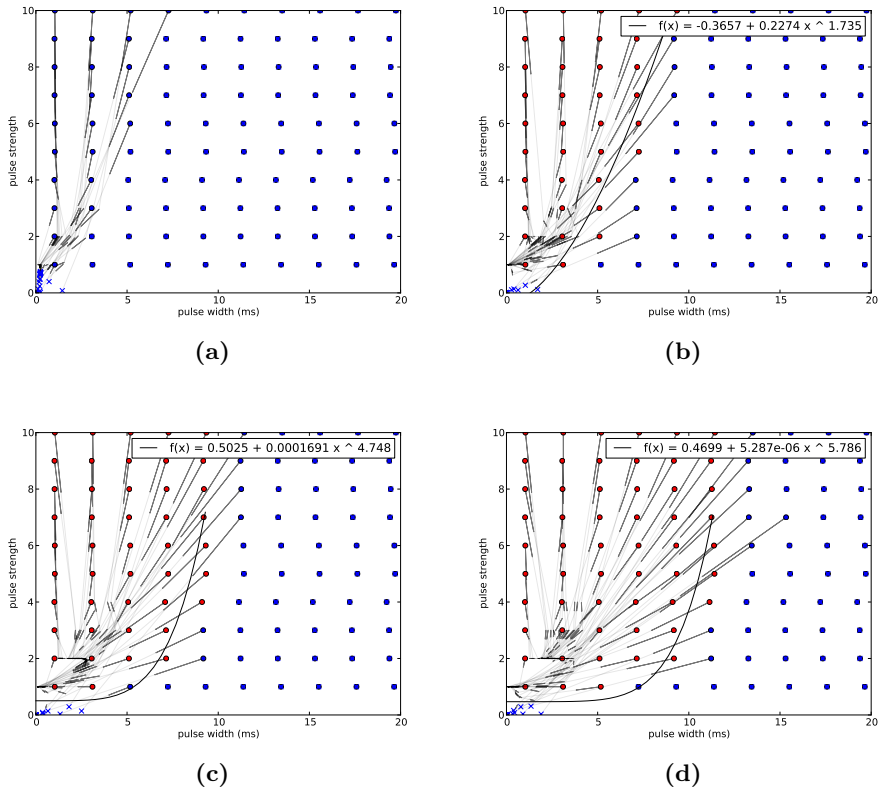


Figure 5.9.: Effect of local delay of the projection between the inhibitory and excitatory population of a synfire group on the filtering properties of the synfire chain. Values of the delay are (a) 0.1 ms (b) 4 ms (c) 6 ms (d) 8 ms. This corresponds to the observations made in [*Kremkow et al.*, 2010, Figure 4]

- The change to the model should act by delaying the effect of a spike on the membrane potential. While this constraint may seem obvious, it is imposed to rule out arbitrary changes that only work in a special case.

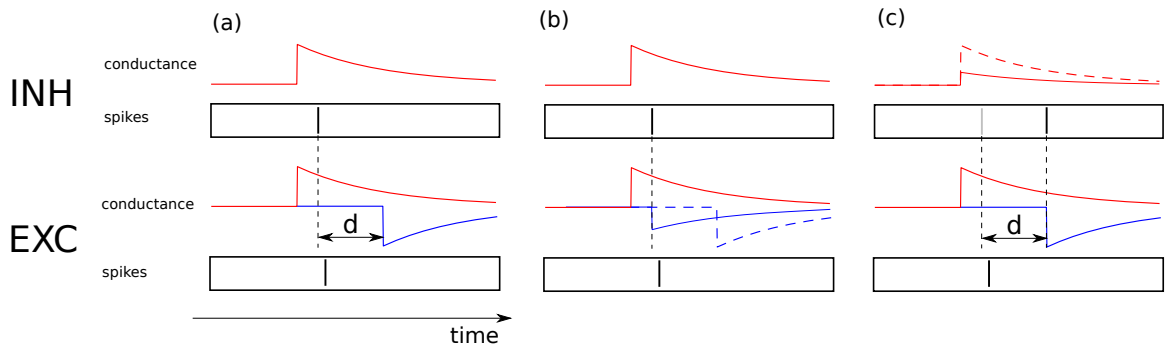


Figure 5.10.: Illustration of possibilities to compensate missing synaptic delays. (a) Original behavior when synaptic delays are present. The conductance course and spike times of an exemplary neuron from an inhibitory (top) and excitatory (bottom) neuron within the same group. The onset of inhibitory conductance is delayed by a time d. (b) No synaptic delay is present. The inhibitory conductance has an increased time constant $\tau_{\text{syn,I}}$ and a reduced weight in comparison to the original conductance course, that is denoted by a dashed line. (c) The spike time of the inhibitory neuron is delayed by a modification of the synaptic weight and time constant from the excitatory neurons of the previous group to the inhibitory neuron.

There are two possibilities to reestablish the original functionality of delays:

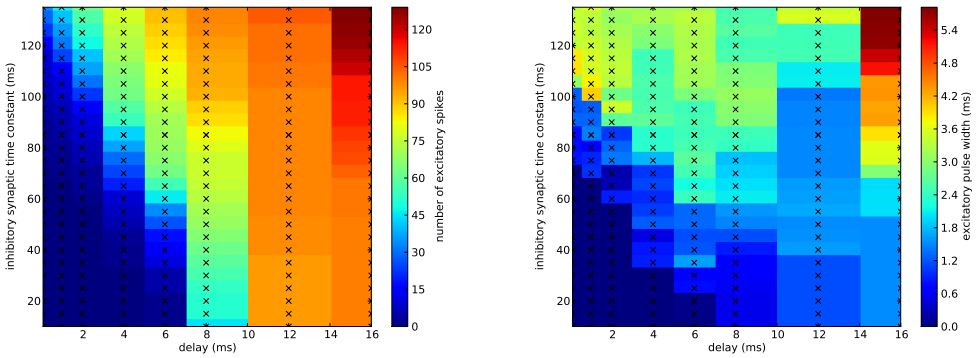
- To slow down the effect of the local inhibition by replacing a delayed rise and fall of inhibitory conductance by an immediate conductance change of a different duration and magnitude. This method is sketched in Figure 5.10 (b).

- To delay the spike time of each inhibitory neuron, ideally keeping the conductance time course seen by the excitatory population exactly the same as in the case of real delays. This spike time delay is accomplished by similar means as in the first option, but now applied to the synapses which make up the intergroup projections from the excitatory to the inhibitory population. (Figure 5.10 (c))

Qualitative compensation by delaying the effect of the inhibition

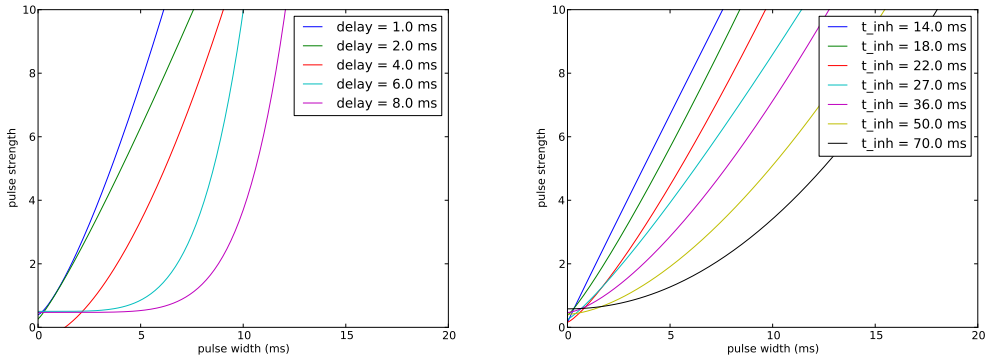
The first compensation variant is implemented by increasing the inhibitory synaptic time constant. To reduce the parameter space from two dimensions (the inhibitory synaptic time constant $\tau_{\text{syn,I}}$ and synaptic weight w) to one, the product $\tau_{\text{syn,I}} \cdot w$ was kept constant.

At first, the question arose, whether such a scaling alone can be enough to replace synaptic delays, even to some degree. The first idea is to see whether a continuous transition is possible, i.e. whether one can slowly decrease the delay while simultaneously increasing the synaptic time constant without changing the synfire behavior. For that, the response of the excitatory population of the first group was taken as a criterion. Figure 5.11 shows this response to a stimulus with $a = 2, \sigma = 10$ ms. While the response strength increases with delay and the



(a) **(b)**

Figure 5.11.: Spiking behavior of the excitatory population of the first group in the chain as a function of the synaptic inhibitory time constant and local delay. The product of the synaptic time constant and the synapse weight was kept constant. The input stimulus had the parameters $a = 2, \sigma = 10$ ms.



(a) **(b)**

Figure 5.12.: Location of the synfire chain separatrix. (a) Real synaptic delays from original model. (b) Scaling of the inhibitory synaptic time constant. The synaptic weight was scaled inversely proportional to the time constant.

inhibitory time constant, the response width does not follow the same pattern. From this alone it is clear that a precise restoration of the original behavior is not possible.

Nevertheless, the location and shape of the stable region can be modified by the given method. Figure 5.12 shows the location of the separatrix in the case of real delays (a) and modification of the inhibitory synaptic time constant (b). While a stable region can be achieved and modified using this method, the shape that is caused by conduction delays of 6 ms and more can not be reproduced. The reason is clear: for large delay values and, accordingly, large synaptic time constants, the difference between the two conductance time courses is largest. A sharp increase after a long delay is replaced by a slowly decreasing conductance, for each spike of the inhibitory population.

Delaying the Spike Time of the Inhibitory Population

This section presents the investigation of the second delay compensation method, as sketched in Figure 5.10 (c).

A first approach roots in the observation that the dynamics that is described by a differential equation

$$\tau \dot{x}(t) = F(x(t), t) \tag{5.2}$$

can be modified to yield the delayed solution x'.

$$x'(t) := x(st) \tag{5.3}$$

The changes affect only the time constant and the explicit time dependency of the right-hand side.

$$s\tau \dot{x}'(t) = F(x'(t), \frac{t}{s}) \tag{5.4}$$

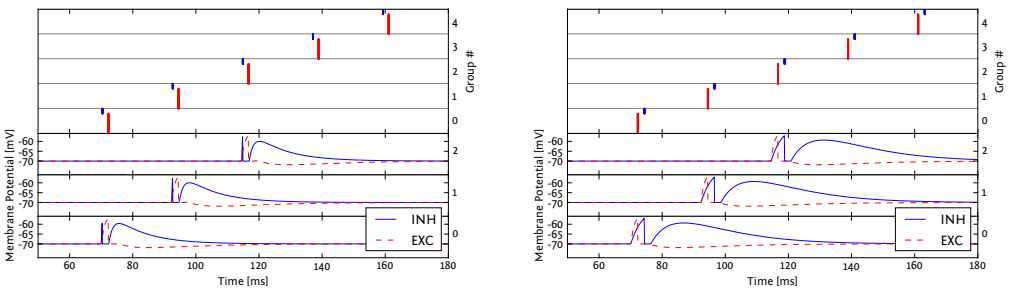
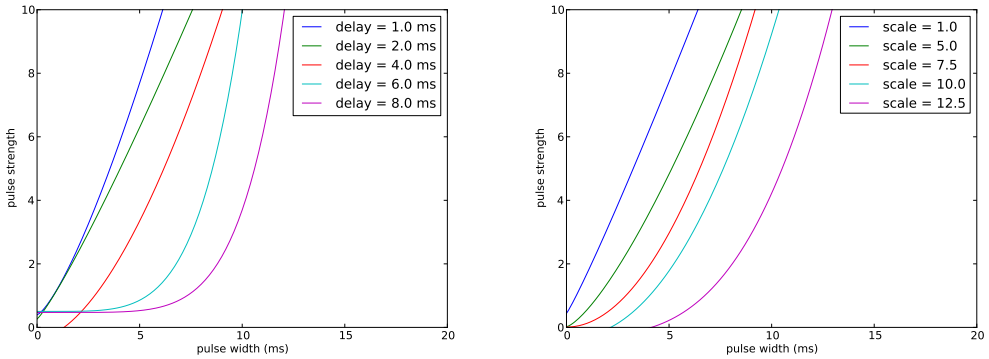


Figure 5.13.: Example for delay compensation by changing the time scale of neuron dynamics. Both figures show a raster plot of the first five groups and exemplary voltage traces taken from the first three groups. (a) Synfire chain without background stimulation and original parameters. (b) Synfire chain with increased synaptic time constant for the inhibitory population (blue, solid curves) and no delay between inhibitory and excitatory population. The volage course of each excitatory neuron (red, dashed curves) stays the same in both cases.



(a) **(b)**

Figure 5.14.: Location of the synfire chain separatrix. (a) Real synaptic delays from original model, identical to Figure 5.12 (a). (b) shift of the separatrix by time scaling of the inhibitory population's membrane and synapse dynamics.

The neuron models employed by the FACETS hardware can be described by equation 5.2; thus, the dynamics can be slowed, and therefore, in a sense, delayed for the synfire chain. Here, the time span between stimulus onset and spike in the inhibitory population is stretched so the spike occurs after 4 ms instead of the fraction of a millisecond in the unmodified model.

Obviously, this kind of modification is very sensitive to the presented stimulus. The amount of delay introduced by this method is not the desired additive, but a multiplicative one. Another obvious drawback is that subthreshold dynamics are equally delayed, thus changing the return time to the resting state by an equal amount. A partial remedy to the second problem is to only scale the synaptic time constant, as shown in Figure 5.13. This abandons the exact computability of the achieved delay, which however is not a major concern in most cases due to the previously mentioned stimulus dependence.

Evaluation of Compensation Methods

The scaling possibilities discussed above have been tested in the default stimulus region. For this, the parameters τ_m, $\tau_{\text{syn,E}}$ and C_m are scaled by a factor. (Note that this employs the time scaling method, as presented in equation 5.4, only up to the first spike, because the duration of the refractory period is not scaled.) In this case, the original current background is employed to avoid an individual adaptation of background stimulus parameters for each scaling factor. As can be seen in Figure 5.14, the separatrix can also be shifted by the presented method; its shape differs naturally from the one caused by real delays. Further deviations are seen in Figure 5.15 (a). The set of stimuli that evokes a response in the first, but not in the last group is bigger when comparing it with 5.9. For a scaling factor of 15, the irregularity of the separatrix is apparent, which is also expected because especially for large σ, fluctuations of the spike distribution within the stimulus packet have a strong influence on the effectively weakened and prolonged effect on the spiking behavior of the inhibitory population.

The quality of the given compensation method for missing delays can be understood better by calculating the *effective delay* of the inhibitory population spike times. It is defined as the

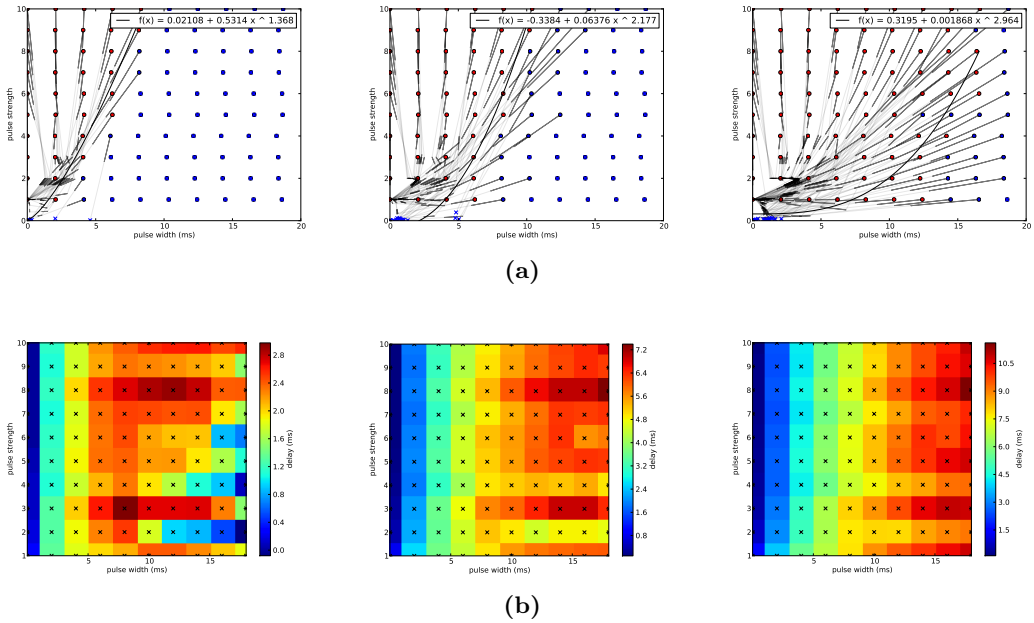


Figure 5.15.: Effect of time scaling of inhibitory neuron's parameters. (a) State space for a scale factor of 5, 10 and 15 (from left to right) (b) *Effective delay* for the same scaling factors as in (a). See text for details.

difference between the mean first spike time in the modified and in the original synfire chain. Only the first spike time is considered, even if a neuron spikes multiple times. T

The visualization of the delay in Figure 5.15 (b) shows the effective delay depending on the location in the σ, a state space. The aforementioned dependence on the stimulus characteristics is shown here. This dependence can be explained easily: for small σ, each neuron sees a rapid conductance rise, and the moderate modifications of the constants do not significantly delay the spike time. For larger σ, the total delay is a result of two effects: the slower rise of an individual EPSP and the increase of the integration time window. The first causes the membrane potential to reach threshold later than in the non-modified case. In contrast, the second can cause an earlier spike when stimulus spikes are sparse. This can be observed in the case of a scale factor of 5 (Figure 5.15 (b), left image): The effective delay increases and then decreases with rising σ, especially for small values of a.

The second point that is visible in Figure 5.15 (b) is the fluctuation of the local delay between neighboring points in the state space. This effect is caused by variations in the concrete realization of the stimulus that is sampled from a Gaussian distribution.

Both inhomogeneities of the local delay, the dependence on σ, a and on the concrete stimulus realization illustrate the limits of a reintroduction of precise synaptic delays.

Establishing the Limits of Effective Delay

The effective delay is now employed to make two quantitative statements about delay reintroduction. The first is the size of the maximally possible effective delay. The second is the spread of effective delay, that relates the mean effective delay to its input-dependent variation.

To establish an upper bound on the possible effective delay, its dependence on three neuron parameters is studied: the membrane time constant τ_{m}, the synaptic time constant $\tau_{\mathrm{syn,E}}$ and the synaptic weight w. The task is to find the maximal value of delay that can be achieved by tuning of these parameters.

The size of the maximal delay could be measured by a sweep over the three parameters. However, the following observation allows for an easy way to reduce the dimensionality: Decreasing each of the parameters leads to a weaker response of the membrane potential, eventually preventing the occurrence of a spike for a certain value of the parameter. This is clear for $\tau_{\mathrm{syn,E}}$ and w, and becomes apparent for τ_{m} if one considers the first differential equation of the leaky integrate-and-fire model in the following form for the case of one excitatory synapse being active at $t = 0$:

$$\dot{U} = -\frac{U - V_{\mathrm{rest}}}{\tau_{\mathrm{m}}} - \frac{w}{C_{\mathrm{m}}} \exp\left(\frac{-t}{\tau_{\mathrm{syn,E}}}\right)(U - V_{\mathrm{rev,E}}) \tag{5.5}$$

For very large τ_{m}, the first addend on the right hand side becomes insignificant. For very small τ_{m}, it becomes dominant, causing the membrane potential to stay near V_{rest}. The maximal effective delay occurs at the border between a single and no spike for each of the three parameters. Thus, scanning this border is enough to find the maximal possible effective delay for a given stimulus.

The regions for the three variables were chosen according to the limits given by the wafer scale neuromorphic system at a speedup of 10^4, given as: $\tau_{\mathrm{m}} \in [1\text{ ms}, 588\text{ ms}]$, $\tau_{\mathrm{syn,E}} \in [1.2\text{ ms}, 86\text{ ms}]$ and $w \in [0.035\text{ nS}, 700\text{ nS}]$ (These values were chosen to be larger than the practically possible hardware values, so any realizable combinations of hardware values lies within these limits.)

Figure 5.16 shows the delay at the border, as defined above, for a temporally sharp (a) and broad (b) stimulus. Both plots show a significant influence of the membrane time constant on the effective delay; The largest effect is observed for a maximally increased membrane time constant (in the given case, by a factor of 10). The reason is, that the source of the delay is the retardation of the spike until most of the stimulus has been applied. This kind only happens if all EPSPs are small but act for a prolonged period. Figure 5.16 (c) shows, how this happens: in the original model, the first spike occurs at the beginning of the stimulus, while in the modified case it occurs after most of the presynaptic neurons have fired. This also explains the much greater effective delay in the case of a large σ, and limits the effective delay to the same order of magnitude as the delay for moderate scaling values. This is consistent with the magnitude of effective delay in the state space, as shown in Figure 5.15.

Having established a limit on the *magnitude* of the maximal possible delay, what remains is to test its *consistency*. The measurement of effective delay in different points of state space has shown, that it varies with the temporal width of the stimulus. To estimate the extent to which this happens at different values of the three modified variables, a final measurement is conducted. A set of points is chosen that covers the default input region: $a \in \{2, 10\}$ and $\sigma \in \{1\text{ ms}, 10\text{ ms}, 15\text{ ms}, 20\text{ ms}\}$. The effective delay is measured in each of those eight points for different values of τ_{m}, $\tau_{\mathrm{syn,E}}$ and w. The mean and *spread* of the delay is measured, the spread being defined as the difference between maximal and minimal delay. Figure 5.17 shows the result: Only a very small region around 0 ms delay has a spread of 0 ms; with increasing effective delay the spread increases by at least the same amount, showing the limited consistency of effective delays for a large number of model modifications.

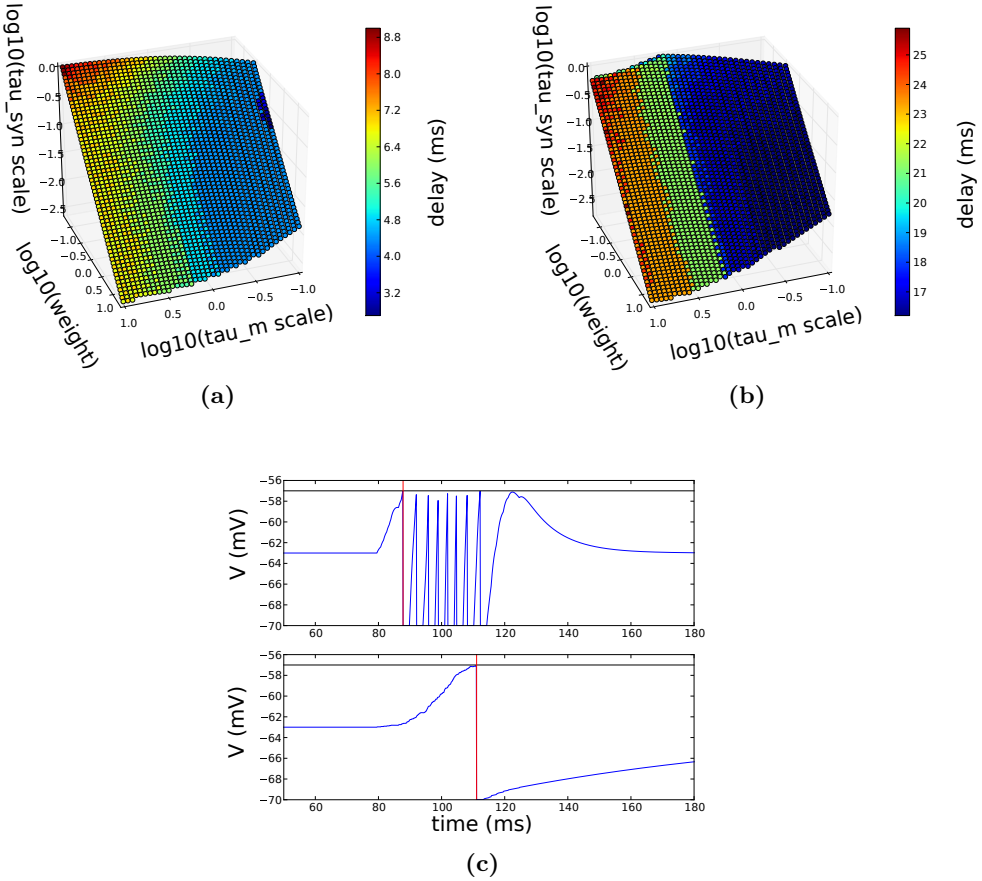


Figure 5.16.: Effective delay of a single inhibitory neuron with modified $\tau_{\mathrm{syn,E}}$, τ_{m} and w at the border between one and zero produced spikes. To minimize trial-by-trial variation, the background stimulus was replaced by a raised membrane potential to -63 mV. The stimulation parameters were (a) $(\sigma, a) = (2\ \mathrm{ms}, 2)$ (b) $(\sigma, a) = (8\ \mathrm{ms}, 4)$ (c) Sample voltage traces for the stimulus given in (b). Stimulus spike times were drawn from a Gaussian distribution with a mean of 100 ms and a standard deviation of 8 ms. Unmodified (upper plot) and modified (lower plot) cases are shown. The modified parameters are: $\tau'_{\mathrm{syn,E}} = 10^{-1.55} \cdot \tau_{\mathrm{syn,E}}$, $\tau'_{\mathrm{m}} = 10 \cdot \tau_{\mathrm{m}}$ and $w' = w$. The red vertical line marks the time of the first spike. The horizontal line denotes the spiking threshold.

5.1.8. Conclusion

In this chapter, the effects of expected hardware distortions on the behavior of the synfire chain model (*Kremkow et al.* [2010]) have been investigated. The effects include correlations in the background stimulus for the network, synaptic weight jitter, synapse loss and unavailability of synaptic delays for Layer 1 communication.

The influence of background correlations has been shown to play a minimal role on the pulse width for correlation values of at least 50%. A distribution of hardware Poisson source generators on a HICANN to neurons in each synfire group can be found that causes a correlation of at most 50%, implying that for a synfire chain model with the utilized parameters, size and interconnection is not significantly affected by background correlations if the available resources are used in an efficient manner.

The effect of weight jitter was likewise minimal, even at values as high as 40%, corresponding to the measurements on an uncalibrated Spikey chip, which have been considered as a worst-case scenario. (See chapter 4)

Synapse loss disrupted stable propagation for a loss probability between 30% and 40%. Compensation by increase of synaptic weights of the remaining synapses led to a reestablishment of stable activity propagation. The increasing sparsity of interconnections led to a change of transmission properties, as was expected considering the results from *Kumar et al.* [2010] with respect to propagation properties of sparse and dense synfire chains.

Compensation of missing delays between the inhibitory and excitatory populations of a synfire group was attempted using two methods.

First, delaying the effect of inhibition on the excitatory population by a modification of inhibitory synaptic time constants and weights caused a shift of the stable region in the (σ, a) state space, which was, however, not comparable in shape to the shift caused by real synaptic delays.

Second, changing the spike time of the inhibitory population by a modification of its time constant, membrane capacitance and the synaptic time constant from the preceding group to the inhibitory population. Similarly to the first method, only an approximate reproduction of the desired behavior was achieved.

The fact that the second method actually shifts a spike time was exploited to define an effective delay that made it possible to investigate its behavior quantitatively. The limits on the magnitude and consistency of possible delays using the second method were established. The result confirmed that the dependence of effective delay on the stimulus shape is so strong that none of the tested combinations of compensation parameters would produce the same effective delay for a variety of presented inputs.

In conclusion, the behavior of the synfire chain network model was established for the given distortions, and possible remedies investigated. The most important result would be that qualitative mimicry of synaptic delays is possible regarding the effect of the separatrix in the (σ, a) state space, given the delay is small (approximately 4 ms in biological time). This complements the delays of at least 5 ms (biological time) available for the Layer 2 communication. (compare section 1.2.2)

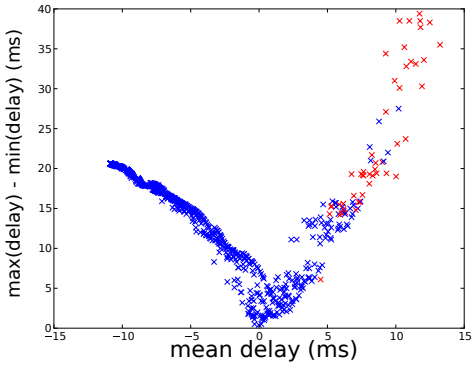


Figure 5.17.: Mean and spread of effective delay for several scaling parameters. The effective delay was calculated using eight points in the state space using $a \in \{2, 10\}$ and $\sigma \in \{1\text{ ms}, 10\text{ ms}, 15\text{ ms}, 20\text{ ms}\}$. Each point represents a distinct combination of $\tau_{\text{syn,E}}$, τ_{m} and w. The variables take on ten values in the regions of $0.1 - 10$-fold of the default value for τ_{m}, $0.016 - 1$ of the default value for $\tau_{\text{syn,E}}$ and $0.05 - 5$ of the default value for w. When stimulation in each point of the (σ, a) space led to a spike, the point is denoted by a blue, otherwise by a red cross.

5.2. Self-Sustained Asynchronous Irregular States

In this part of the thesis, a set of randomly connected networks with complex neuron behavior are investigated with respect to their reaction to applied distortions and compensation.

Motivation

Cortical neurons show irregular activity in awake mammals (*Destexhe et al.* [2003], *Lee et al.* [2006]) that can be described as "asynchronous irregular" (AI) states (*Destexhe* [2009]). Such states can be observed in large networks of current-based (*Brunel* [2000]) and conductance-based (*Vogels and Abbott* [2005]) integrate-and-fire neurons. Neurons in thalamus and cortex exhibit more complex firing patterns than the ones that can be reproduced by simple LIF neurons. The work presented in *Destexhe* [2009] examines the occurrence of AI states in networks of neurons that display such complex behavior. It is simulated by exploiting the versatility of the *AdEx* model to mimic firing patterns that are observed in the mammalian cortex.

The investigation of this benchmark model is worthwhile for several reasons. The use of the adaptive exponential allows to use the capabilities of the HICANN module on the wafer scale neuromorphic system. A completely random connectivity enables a test for the efficiency of the mapping and routing algorithms. The limits of hardware variations that affect the viability of a neuroscientifically relevant simulation are examined.

5.2.1. Network Model Definitions

Destexhe [2009] uses the Adaptive Exponential neuron model to replicate several types of common cortical and thalamic neurons. Cortical layers consist of regular spiking (RS) cells that show spike-frequency adaptation. The magnitude of the adaptation is varied using different values for the adaptation variable increment b. Fast spiking cells are inhibitory cells that are modeled without adaptation. The influence of rebound spiking neurons on the network stability is also investigated. This is accomplished by introducing low threshold spiking (LTS) cells that show spike frequency adaptation and rebound bursts. The RS and LTS celltypes are pyramidal (PY) cells. Thalamic layers consist of thalamocortical (TC) excitatory and thalamic reticular (RE) inhibitory neurons, both of which show rebound bursts and moderate (TC) resp. strong (RE) adaptation. The complete neuron parameters are listed in A.3.1.

The following network models are presented in *Destexhe* [2009] and were implemented using *PyNN*.

Thalamic Network

Small circuits of thalamic neurons show self-sustained oscillations (*Timofeev and Bazhenov* [2005]). A simple network consisting of 50% TC and 50% RE neurons is constructed at different sizes, with connection probabilities of 2% TC $\rightarrow$ RE, 8% RE $\rightarrow$ TC and RE $\rightarrow$ RE. There are no connections within the excitatory TC populations. The connection probabilities are given for a network of 100 TC and RE cells each; for other sizes, they are rescaled to keep the mean number of incoming connections constant. All excitatory synaptic weights in all networks equal 6 nS, all inhibitory weights 67 nS.

The rebound properties of TC and RE cells ensure a sustained activity even at small network sizes.

Single Layer Cortical Network

A network that consists of 20% inhibitory FS neurons and 80% excitatory neurons. The excitatory neurons are mostly RS cells with a small (0% to 20% depending on the setup) proportion of LTS cells. The connection probabilities are given for a total neuron count of 2000 and rescaled to preserve the mean number of incoming connections.

At a network size of 2000 neurons, even with 0% LTS cells, activity can be sustained for several seconds. For smaller networks (of about 500 cells), LTS cells counteract a fast termination of activity by rebound bursts.

Thalamocortical Network

A network that connects a thalamic and a single layer cortical network with their default sizes. The excitatory neurons of each layer connect with a probability of 2% to all neurons of the other layer, inhibitory connections are only local. These connection probabilities are likewise rescaled to ensure a constant mean fan-in.

In this case, the thalamic layer ensures a persisting activity in the much larger cortical layer. Depending on the amount of adaptation in the cortical RS neurons, an active state with a continuously high firing rate (weak adaptation) or a repeating transitions between UP and DOWN states.

Weak adaptation is realized by setting the value of b to 5 pA for RS cells, and strong adaptation by setting b to 20 pA.

Two Layer Cortical Network

This is a network consisting of a small cortical layer with LTS cells that can sustain activity and a large layer without LTS cells. In analogy to the thalamocortical network, the small layer ensures sustaining activity in itself and in the larger layer. The larger layer is comprised of 2000 neurons and the smaller of 500 neurons, with 400 Pyramidal cells of which 10% are LTS. The connection probability of an excitatory neuron of one layer to any neuron of the other layer is 1%.

The reason for the setup, in addition to the thalamocortical network, is the observation of self-sustained activity and UP/DOWN states in cortical slices (*Sanchez-Vives and McCormick* [2000]).

Initial Stimulus

All networks are only stimulated in the beginning of the experiment for 50 ms. Up to 20% of the network's population were stimulated a firing rate of 200 Hz to 600 Hz, after which self sustained activity begins.

5.2.2. Functionality Measures

In *Destexhe* [2009], the main characteristics that are used to classify network activity states are the correlation coefficient (CC), the coefficient of variation of interspike intervals (CV_{ISI}),

and the mean firing rate. The correlation coefficient measures the synchrony between different neurons in the network. The coefficient of variation shows the irregularity of spike patterns. These quantities are common to characterize activity states in simulations (*Brunel* [2000]), (*Kumar et al.* [2008]) and in-vivo recordings (*Shinomoto et al.* [2005]).

Correlation Coefficient

The correlation coefficient is defined as the averaged cross-correlation of pairs of time-binned spiketrains S_i.

$$\mathrm{CC} = \left\langle \frac{\mathrm{Cov}(S_i, S_j)}{\sigma(S_i)\sigma(S_j)} \right\rangle \tag{5.6}$$

The mean is taken over at least 200 disjoint pairs of spiketrains, with a time bin of 20 ms. Cov denotes the covariance of the two spiketrains and σ the standard deviation. This measure quantifies the amount of synchrony present in the network.

Coefficient of Variation of Interspike Intervals

The coefficient of variation is defined as the mean relative variation of interspike intervals:

$$\mathrm{CV}_{\mathrm{ISI}} = \left\langle \frac{\sigma_i^{\mathrm{ISI}}}{\overline{\mathrm{ISI}_i}} \right\rangle \tag{5.7}$$

$\overline{\mathrm{ISI}_i}$ and σ_i^{ISI} denote the mean and standard deviation of the interspike intervals in spiketrain i. The average $\langle\rangle$ runs over all spiketrains i in the network. Thus, $\mathrm{CV}_{\mathrm{ISI}}$ quantifies the amount of irregularity in the spiking behavior. For example, a completely regular spiking pattern has $\mathrm{CV}_{\mathrm{ISI}} = 0$, a Poisson spiketrain has $\mathrm{CV}_{\mathrm{ISI}} = 1$.

5.2.3. Local Variation

While these criteria are appropriate as a concise description of the activity state, more detail is desired for the investigation within the scope of this thesis. As the main question is whether, and to what extent the expected hardware imperfections will cause altered network activity, it makes sense to consider further state measures.

Because of the often occurring bursting behavior, and the fact that $\mathrm{CV}_{\mathrm{ISI}}$ is not enough to distinguish irregular from periodic bursting behavior, a further measure of regularity was introduced. *Shinomoto et al.* [2005] defines Lv, the coefficient of *local variation* as follows:

$$\mathrm{Lv} = \frac{1}{n-1} \sum_{k=1}^{n-1} \frac{3(\mathrm{ISI}_k - \mathrm{ISI}_{k+1})^2}{(\mathrm{ISI}_k + \mathrm{ISI}_{k+1})^2} \tag{5.8}$$

Lv is normed such that, in analogy to CV, it takes a value of 0 for non-varying interspike intervals and a value of 1 for a Poisson spike train. Contrary to CV, it depends on the local heterogeneity of the spiking pattern.

Characterization of Network States

The term "asynchronous irregular" is defined by *Destexhe* [2009] using the two measures CC and CV_{ISI}. A low CC of < 0.1 means, the spiking behavior is "asynchronous". When $CV_{ISI} > 1$, the spiking behavior is considered "irregular".

5.2.4. Influence of Distortions and Model scaling

The influence of distortions on the models is investigated to establish their intrinsic stability. Because synaptic delays are not incorporated in the networks, only the influence of synaptic weight jitter and synapse loss need to be considered.

Synaptic Weight Jitter

Because of the random connectivity of all four network models considered in this section, weight jitter is not expected to have a strong influence on the performance of any network model. The effects are shown in section A.1 in full. An example for the two-layer cortical network is shown in Figure 5.18.

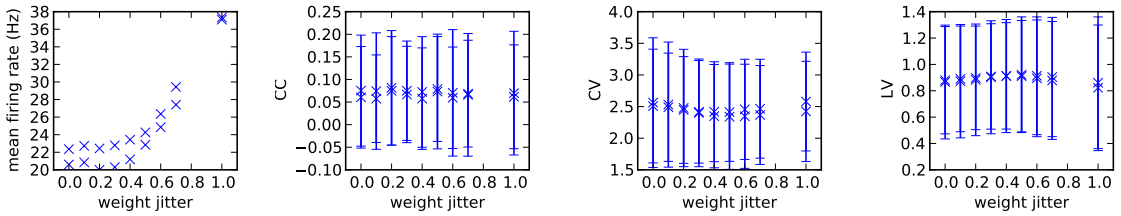


Figure 5.18.: Effects of weight jitter loss on two-layer cortical network in the case of strong adaptation.

The most prominent effect is the increase of the firing rate starting at mean value of jitter of 30% and 50%. The most likely explanation is that for large jitter values, the mean value of the weight distribution increases, because the connection weights are sampled from a Gaussian distribution with the original weight as mean and the product of original mean and jitter value as standard deviation. Negative synaptic weights are then set to 0 (because *PyNN* does not allow negative synaptic weights) which shifts the mean of the distribution for larger jitter values.

The quantities CC, CV and Lv do not, overall, show strong change with varying weight jitter, particularly not in the relevant region < 40 %. There are some outliers for the single layer and two layer cortical networks, where CV increases and has a larger deviation. This is a consequence of a network changing from one stable state to another – an example is shown in Figure 5.19. Because Lv only considers local interspike interval changes, it is not strongly affected.

Synapse Loss: Effects and Compensation

The most obvious effect of synapse loss is the disappearance of self-sustained activity. One possible compensation for synapse loss is a change of all synaptic weights, keeping the product of weight and number of remaining synapses constant. The results of synapse loss and

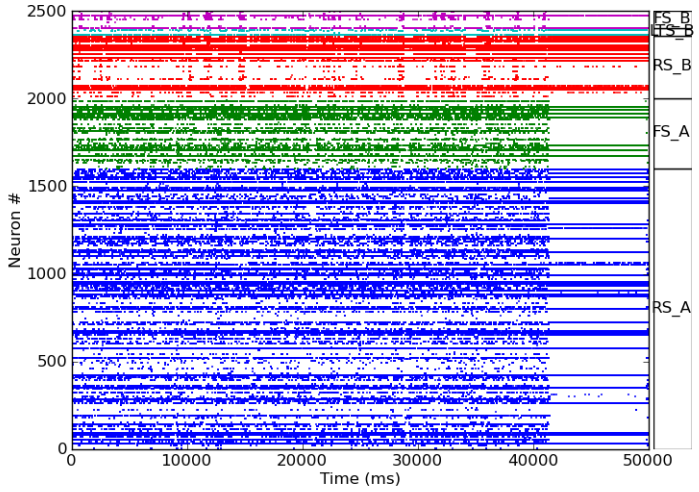


Figure 5.19.: Raster plot of a two-layer cortical experiment that switches between two states at approximately 42 seconds. The experiment was conducted for a weight jitter value of 10%.

compensation are shown in section A.2 in full. An example for the two-layer cortical network is shown in Figure 5.20.

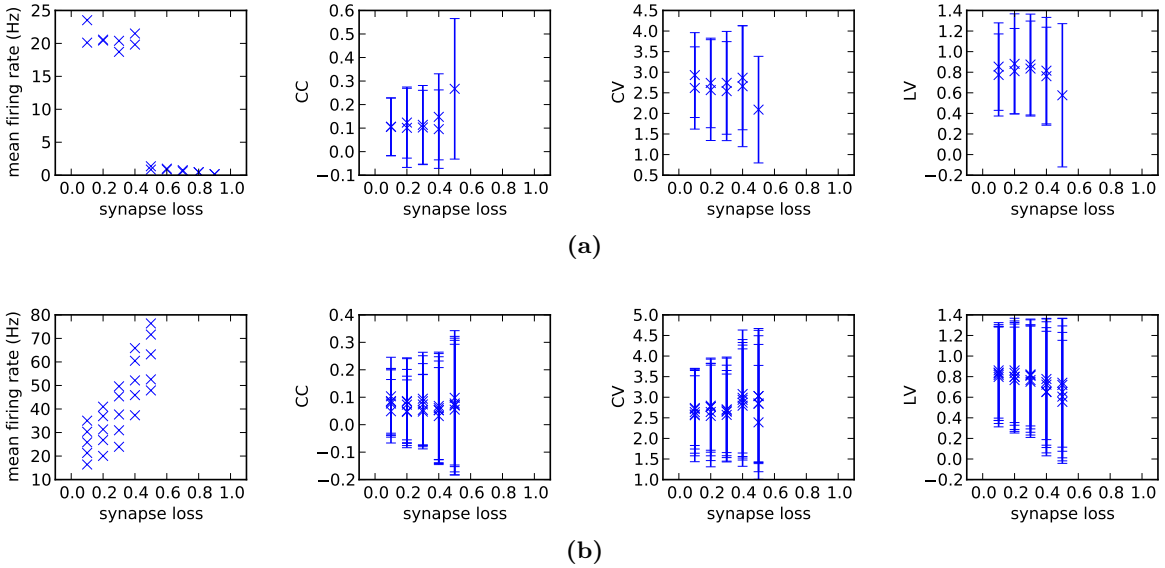


Figure 5.20.: Effects of synapse loss on two-layer cortical network in the case of strong adaptation. (a) No compensation. (b) Compensation by weight scaling.

The activity stops being self-sustained at 50% to 70% synapse loss in all networks. Without compensation, the mean firing rate in each network stays constant or increases up to the point where no stable active state is present. Because both excitatory and inhibitory neurons with

different parameters and complex spiking behavior are present in the network, and both are affected by synapse loss, the result is not obvious to predict.

In the large two-layer networks, CC and its standard deviation generally increase with the value of synapse loss, indicating an increase in both correlation and anti-correlation. This is to be expected, because as the number of inputs of each neuron decrease, the correlation between the remaining connections increases. In the case of inhibitory connections, the expected result would be a stronger anti-correlation. For the smaller networks, no clear trend is evident. In the case of the thalamic network, no CC was calculated because the number of neurons was not large enough. For the single-layer cortical network, the trial-by-trial variation for each of the four measures was higher than any recognizable trend.

In all but the single layer cortical network, a decrease of CV and Lv of different magnitude was observed. Thus, not only do networks synchronize, but they also become more regular in their firing pattern at large values of synapse loss. The exact values at which this happens depend on the respective network model.

When standard compensation is attempted, in which the weight is scaled inversely to the number of remaining synapses, the mean firing rate increases with increasing synapse loss until self-sustained activity becomes completely suppressed and the network activity dies out.The drop begins at 70% for the thalamic network and at more than 90% for the single layer cortical network. In all but the single layer cortical network, which had a large trial-by-trial variation, the standard deviation of CC reliably increases, and the mean Lv decreases. This happens accordingly to expectations, because reducing the number of connections while increasing their synaptic weight should increase the influence of one neuron on another, thus increasing the correlation of their firing patterns. CV does not show a clear trend over all networks.

Scaling of the Network Models

The usefulness of the network models presented in this chapter would be greatly increased if the size of the models was variable, so an assembly of neuromorphic hardware devices of arbitrary size can be tested. In section A.3, the behavior of the four measures is shown for each network at different network sizes. All populations in each network were scaled proportionally to the total size; all connection probabilities between populations were scaled inversely, keeping the mean number of inputs for each neuron constant. As can be seen in section A.3, all measures change only weakly at network sizes larger than 10000 neurons, indicating that at this size, the inputs of each neuron are approximately independent. This means that each of these networks constitutes a stable hardware benchmark at large sizes, to an extent that benchmark results at different hardware sizes become comparable.

5.2.5. Conclusions and Outlook

The work of this part of the thesis concerned the implementation, or re-implementation of four network models that show self-sustained asynchronous-irregular behavior together with a set of analysis functions.

The behavior of the networks in the presence of synapse loss was examined, establishing a negligible effect of synaptic weight jitter at values less than 40%, which is the worst-case scenario considered in this context as a result from the measurements performed on an uncalibrated Spikey chip (Chapter 4).

All networks maintain self-sustained activity up to at least 50% synapse loss. Extending activity past the point of breakdown by scaling all synaptic weights inversely to the number of remaining synapses succeeds, but only at the price of changing the behavior of the network significantly, making this form of compensation rather unfeasible.

The introduction of Lv as additional metric has been beneficial, as it focuses on a different aspect of regularity than Cv, e.g. by not being affected by switches between semi-stable network states while still reflecting the amount of regularity during the simulation.

Simulations of a simple scaling rule of keeping a constant mean number of incoming connections per neuron showed a stable behavior for all 4 network models, making them candidates for the effects of mapping large scale networks on the wafer-based hardware system. However, if this is to be done, a more involved analysis of the respective network will be necessary.

Conclusions and Outlook

The goal of this thesis was an investigation of the influence of hardware distortions on the behavior of different neural network models.

The examined distortions included *synaptic weight jitter*, the variability of synaptic weights caused by hardware production irregularities and digitalization of synaptic weights, *synapse loss*, the result of non-realization of synaptic connections between two neurons when the required connectivity exceeds the available communication resources and the *limited availability of delays* for Layer 1 communication.

Two different approaches were taken to obtain a more complete view of the given problem.

From the experimental side, variability of synaptic weights was measured on the chip-based neuromorphic hardware system.

From the theoretical side, two of the Demonstrator benchmark models were analyzed with regard to their behavior and its change in the presence of idealized distortions. In cases where those distortions greatly affected network behavior, methods were considered to compensate the effect and restore the original behavior by modifications of the model itself. These analyses were conducted via software simulation of the respective models.

The measurement of synaptic weight variability on the chip-based system was conducted due to the need for an expressive quantity for the strength of effective weight jitter. A set of spike-based methods for this measurement was considered, and a best candidate chosen that produces a quantity that can be used directly in the software simulation. Thus, this investigation produced not only a numerical value as a basis for further analysis, but also a method for its acquisition that has been demonstrated to work on a neuromorphic hardware system. Due to the fact that the method is purely rate-based, it may provide a basis for an efficient synapse calibration or a synapse calibration cross-check.

The first of the two analyzed models, namely the synfire chain with feedforward inhibition can be viewed as a representative of a broader class of modular models with purely feedforward transmission and delay-based computation. With a suitable parameter set, the functionality of the model has been shown to remain unaffected by synaptic weight jitter within the relevant boundaries. Synapse loss becomes critical at high values, suppressing signal propagation, but can be easily compensated by increasing the synaptic weights of the remaining synapses. It is important to note that this is only possible due to modularity of the network, which defines its functionality over entire groups rather than individual neurons. The maybe most complex effects are caused by turning off synaptic delays. Compensation by modifying neural and synaptic parameters, especially time constants, does show promising results for reproducing small delays, but displays only limited feasibility in the high delay regime, as these modifications strongly affect signal transmission properties. However this method is perfectly complemented by a feature of the hardware which allows rerouting of synaptic connections through the Layer 2 communication network, which is prohibitive for small delays, but works reliably for large ones.

The second model, which is actually a set of four networks exhibiting self-sustained asynchronous irregular activity, has its functionality defined by abstract measures, such as the

cross-correlation and the coefficient of variation of spike trains. This model showcases the versatility of the hardware neuron implementation, since both the adaptation mechanism and the soft threshold are essential for the functionality of the network. Since the original network model features no delay mechanisms, the only relevant distortions remain the ones caused by synaptic jitter and synapse loss. All four submodels have been shown to tolerate fairly high amounts of both synaptic jitter and synapse loss, certainly more than is expected to occur on the waferscale device, after which the activity stops being self-sustained. An attempt at using the same synapse loss compensation technique of weight scaling that works well for a synfire chain led to an even stronger behavioral change. Future work in this direction might include changing the AdEx parameters of the different cell types.

Altogether, the presented model investigations show that the search for compensation mechanisms provides valuable insights, independent of its success, providing either a feasible strategy to counteract distortions and its limits and drawbacks, or preventing unnecessary work.

The need for expressive performance metrics for each model has been found to be of great importance. While these measures are often provided either by the source of the model itself or the neuroscientific research community, they have to be carefully considered with regard to the effects one may encounter during the use of a neuromorphic hardware system.

It has to be stressed that the precise numerical results of distortion effects on the Demonstrator models depend strongly on the particular model parameters and constitute in no way a final objective, as they can only provide an order of magnitude for these effects. They are, instead, a tool to provide users of neuromorphic hardware with the necessary intuition regarding effects that have to be expected for different types of networks and which hardware-specific effects are expected to be significant. Likewise, the compensation methods presented in this thesis should not only show the exact influence on the concrete model, but also give future hardware users a possibility to better judge a course of action if they encounter hardware distortions during their work with network models that are similar to the ones presented in this thesis.

A very important point has to be raised here: along with other efforts aiming in the same direction, the present work shows how the implemented hardware design and especially its unique versatility concerning the permitted range of network architecture and parameters allows the emulation of a vast panoply of network models, and where the limitations begein having a measurable effect, a variety of compensation measures can usually be found. From this point of view, the FACETS waferscale device does indeed fulfill the necessary features required from a universal modeling back-end.

The continuation of the work presented in this thesis needs to compare the idealized model distortions with more realistic scenarios. For example, comparisons with simulations using the Executable System Specification can show to what extent the idealization of completely random synapse loss made in this theses applies to different network architectures, because the *ESS* model of the mapping and routing process corresponds exactly to the ones for the hardware system and thus, the synapse loss distribution is realistic.

Concerning the future integration of modeling and hardware development, efforts need to be made to prevent a parallel, independent work. With increasing complexity of benchmark models, the strengths and weaknesses of the neuromorphic hardware systems need to be considered continuously during development. Otherwise, the end results risk to be models so complex that hardware-induced distortion compensation might become an impossible task.

Considering that the results of model modifications are not obvious even in simple cases, as shown by the example of synaptic weight scaling for randomly connected networks in

section 5.2.4, the optimization of a given benchmark model that was only developed in a software environment is likely to prove difficult. It is imperative that the hardware (and, until its completion, the ESS) is employed as more than just as a final stage for running well-established models; it really needs to be used as the universal modeling tool it was designed to be in the first place.

A. Self-Sustained Asynchronous Irregular States: Additional Figures

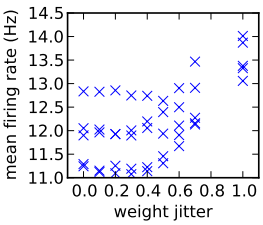
Diagrams of the network analysis presented in 5.2.

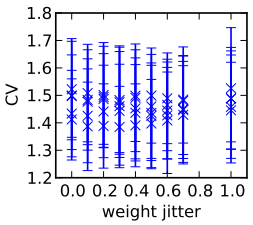
Multiple data points stem from simulations with the same parameters but different random seeds for initial stimulus. Error bars in graphs for CC, CV and Lv denote the standard deviation of each quantity over the sample in which it was calculated.

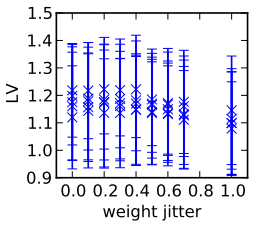
The graphs for the thalamic network do not contain CC as a metric because the network was smaller than the minimum sample size for the cross-correlation average.

A.1. Weight Jitter

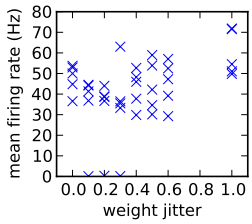
Thalamic Network

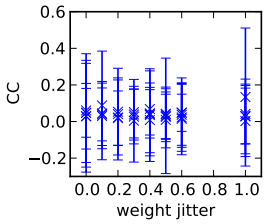


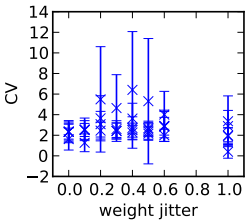


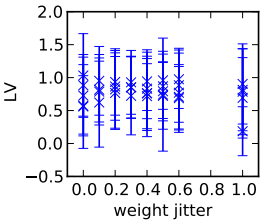


Single Layer Cortical Network

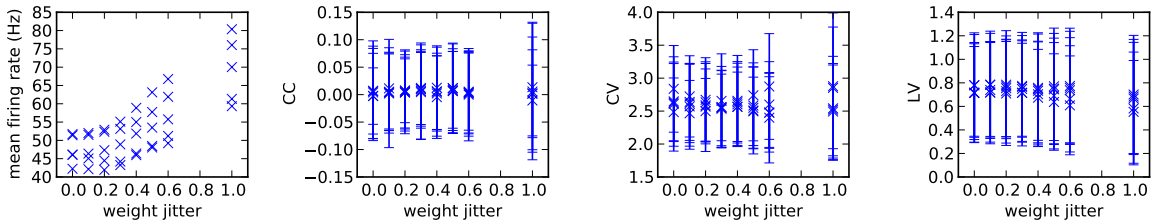




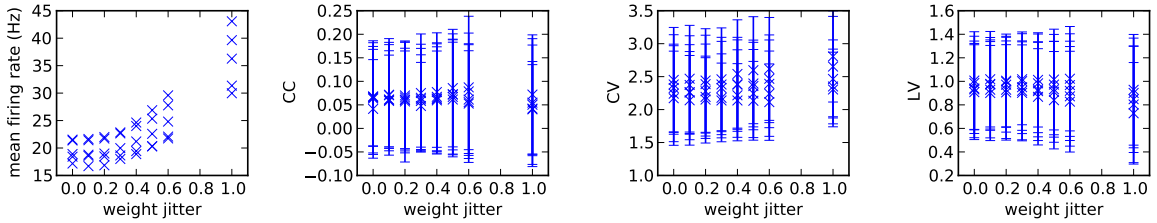




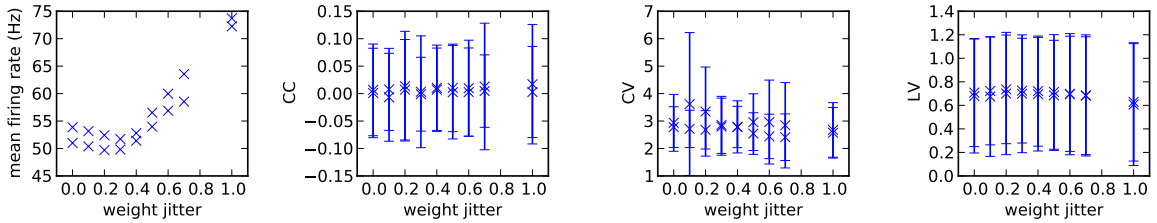
Thalamocortical Network, weak adaptation (b = 5 pA)



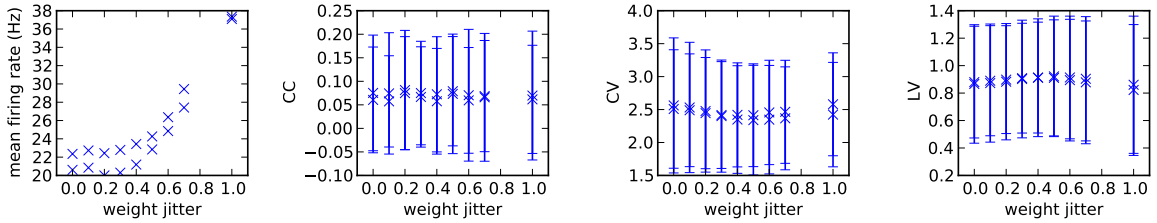
Thalamocortical Network, storng adaptation (b = 20 pA)



Two Layer Cortical Network, weak adaptation (b = 5 pA)

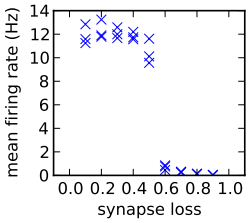


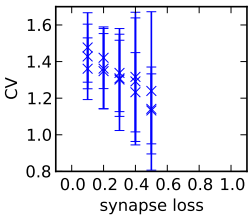
Two Layer Cortical Network, strong adaptation (b = 20 pA)

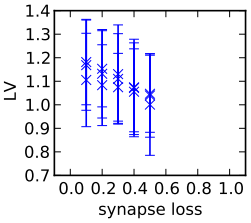


A.2. Synapse Loss

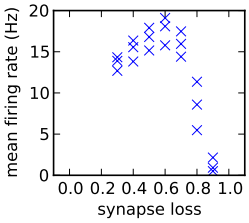
Thalamic Network

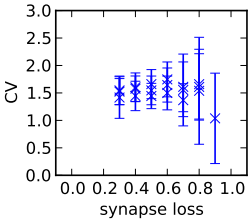


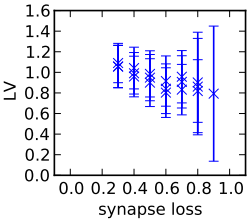




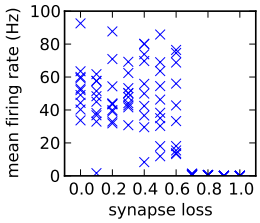
Thalamic Network, compensated

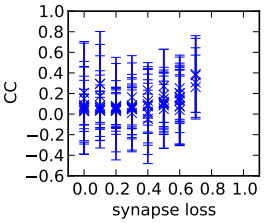


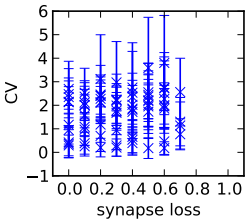


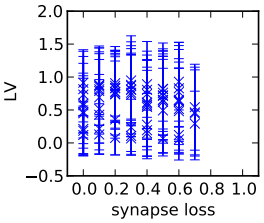


Single Layer Cortical Network

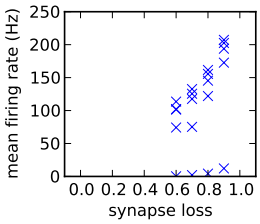


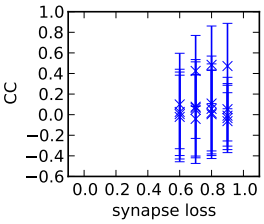


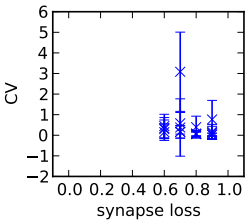


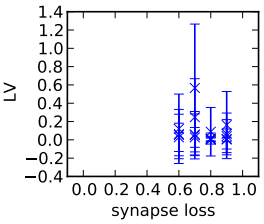


Single Layer Cortical Network, compensated

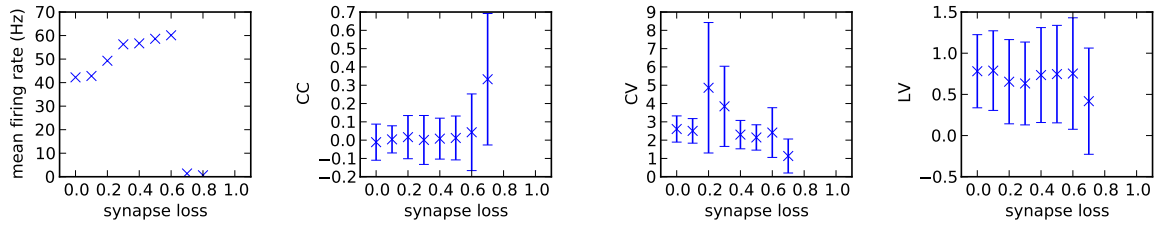




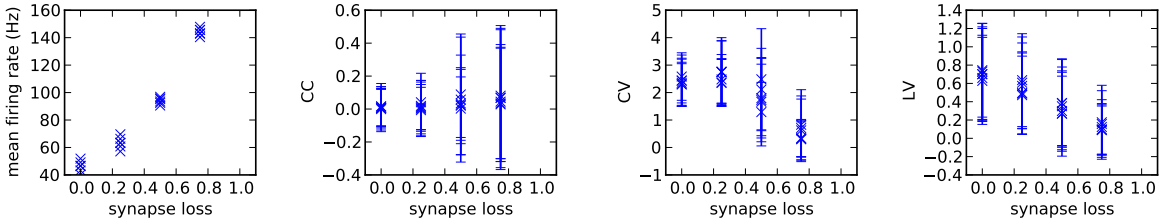




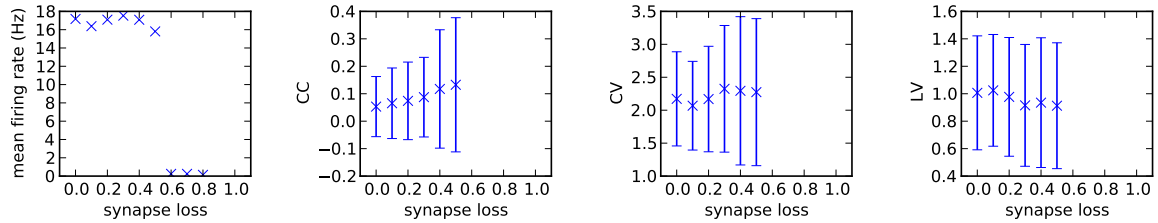
Thalamocortical Network, weak adaptation (b = 5 pA)



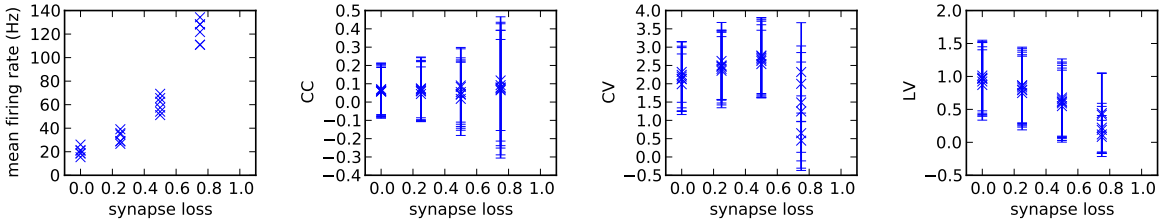
Thalamocortical Network, weak adaptation (b = 5 pA), compensated



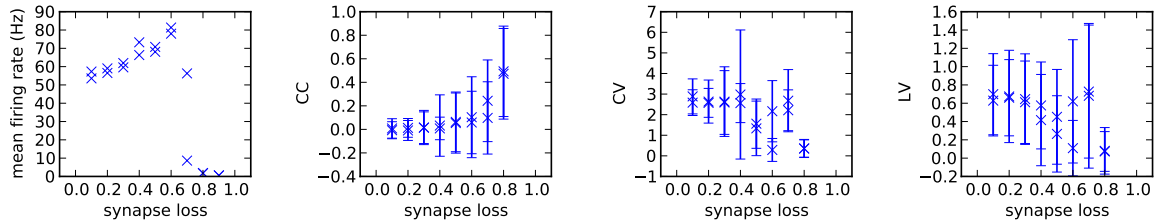
Thalamocortical Network, storng adaptation (b = 20 pA)



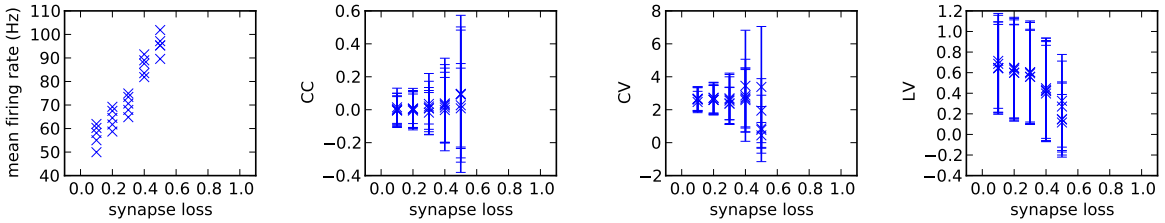
Thalamocortical Network, storng adaptation (b = 20 pA), compensated



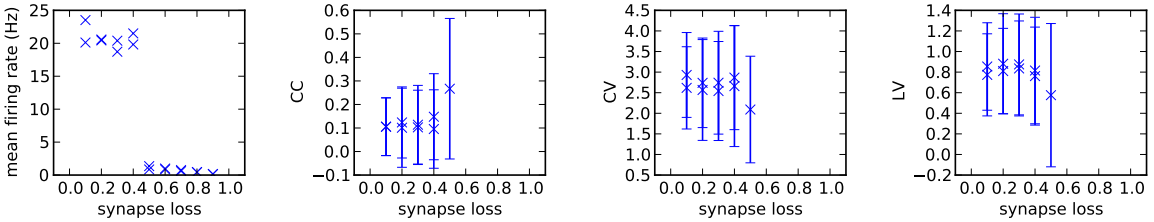
Two Layer Cortical Network, weak adaptation (b = 5 pA)



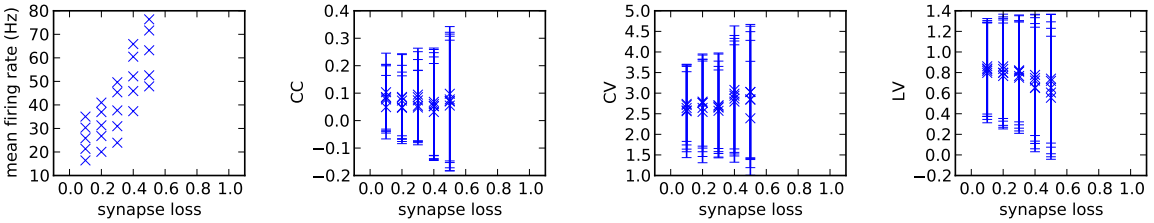
Two Layer Cortical Network, weak adaptation (b = 5 pA), compensated



Two Layer Cortical Network, strong adaptation (b = 20 pA)

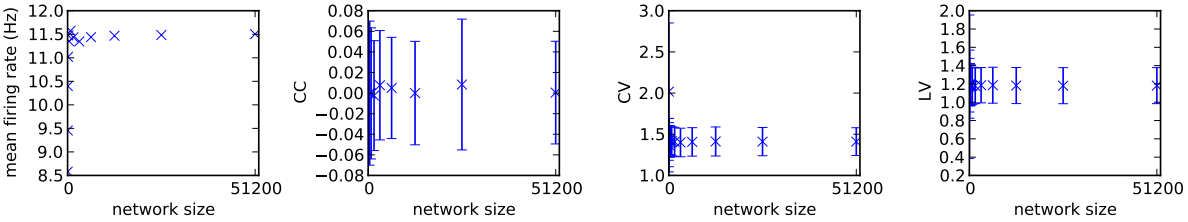


Two Layer Cortical Network, strong adaptation (b = 20 pA), compensated

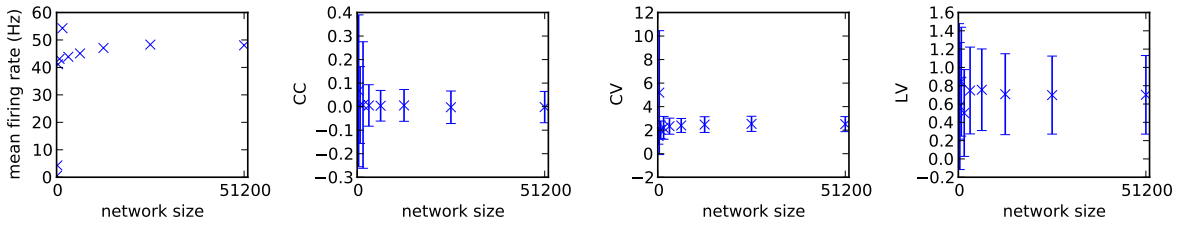


A.3. Network Scaling

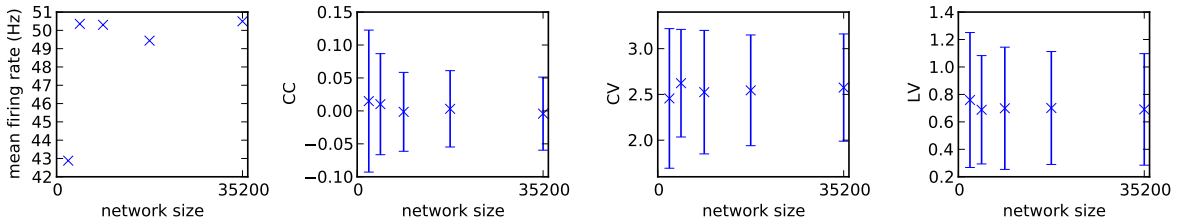
Thalamic Network



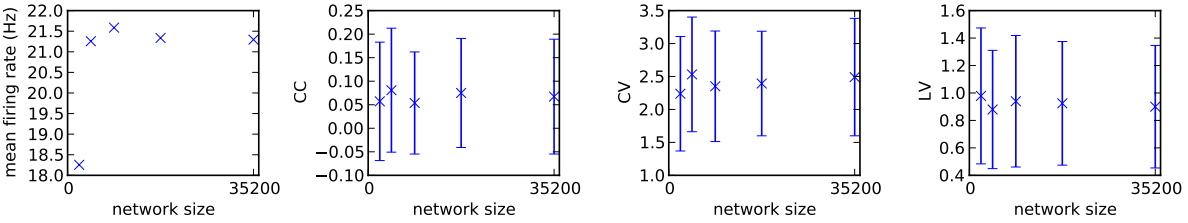
Single Layer Cortical Network



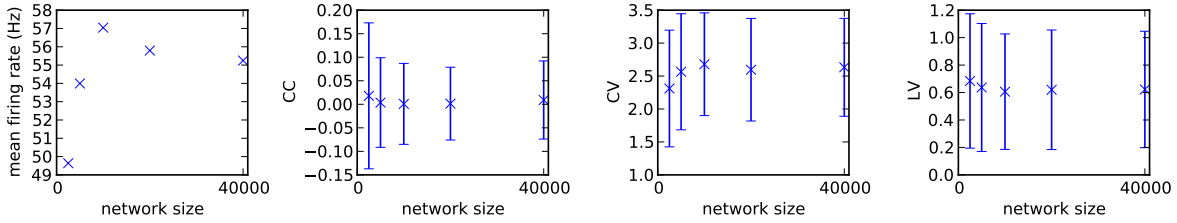
Thalamocortical Network, weak adaptation (b = 5 pA)



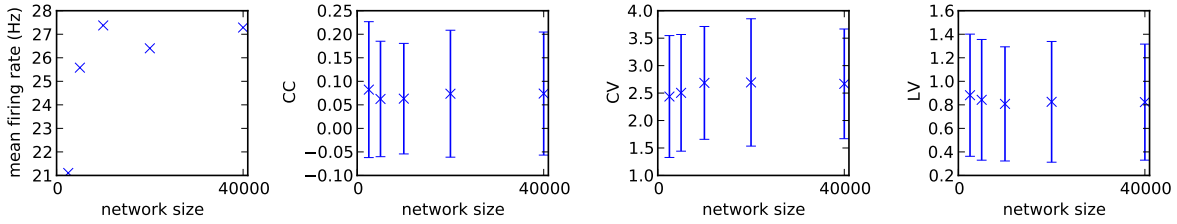
Thalamocortical Network, storng adaptation (b = 20 pA)



Two Layer Cortical Network, weak adaptation (b = 5 pA)



Two Layer Cortical Network, strong adaptation (b = 20 pA)



A.3.1. Neuron Parameters

Neuron parameters for the networks described in section 5.2.1. The values are given in *PyNN* format, i.e. the parameter names and unit values conform to *PyNN* (0.6) standard.

TC

```
{'a': 40.0,
 'b': 0.0,
 'cm': 0.2,
 'delta_T': 2.5,
 'e_rev_E': 0.0,
 'e_rev_I': -80.0,
 'tau_m': 20.0,
 'tau_refrac': 2.5,
 'tau_syn_E': 5.0,
 'tau_syn_I': 10.0,
 'tau_w': 600.0,
 'v_init': -60,
 'v_reset': -60.0,
 'v_rest': -60.0,
 'v_spike': -50.0,
 'v_thresh': -50.0}
```

RE (strong adaptation)

```
{'a': 1.0,
 'b': 0.02,
 'cm': 0.2,
 'delta_T': 2.5,
 'e_rev_E': 0.0,
 'e_rev_I': -80.0,
 'tau_m': 20.0,
 'tau_refrac': 2.5,
 'tau_syn_E': 5.0,
 'tau_syn_I': 10.0,
 'tau_w': 600.0,
 'v_init': -60,
 'v_reset': -60.0,
 'v_rest': -60.0,
 'v_spike': -50.0,
 'v_thresh': -50.0}
```

For "weak adaptation", a value of $b = 0.005$ was used.

RS (strong adaptation)

```
{'a': 1.0,
 'b': 0.02,
```

```
 'cm': 0.2,
 'delta_T': 2.5,
 'e_rev_E': 0.0,
 'e_rev_I': -80.0,
 'tau_m': 20.0,
 'tau_refrac': 2.5,
 'tau_syn_E': 5.0,
 'tau_syn_I': 10.0,
 'tau_w': 600.0,
 'v_init': -60,
 'v_reset': -60.0,
 'v_rest': -60.0,
 'v_spike': -50.0,
 'v_thresh': -50.0}
```

For "weak adaptation", a value of $b = 0.005$ was used.

FS

```
{'a': 1.0,
 'b': 0.0,
 'cm': 0.2,
 'delta_T': 2.5,
 'e_rev_E': 0.0,
 'e_rev_I': -80.0,
 'tau_m': 20.0,
 'tau_refrac': 2.5,
 'tau_syn_E': 5.0,
 'tau_syn_I': 10.0,
 'tau_w': 600.0,
 'v_init': -60,
 'v_reset': -60.0,
 'v_rest': -60.0,
 'v_spike': -50.0,
 'v_thresh': -50.0}
```

LTS

```
{'a': 20.0,
 'b': 0.0,
 'cm': 0.2,
 'delta_T': 2.5,
 'e_rev_E': 0.0,
 'e_rev_I': -80.0,
 'tau_m': 20.0,
 'tau_refrac': 2.5,
 'tau_syn_E': 5.0,
 'tau_syn_I': 10.0,
 'tau_w': 600.0,
```

```
'v_init': -60,
'v_reset': -60.0,
'v_rest': -60.0,
'v_spike': -50.0,
'v_thresh': -50.0}
```

Bibliography

Abeles, M., *Corticonics: Neural Circuits of the Cerebral Cortex*, Cambridge University Press, New York, 1991.

Abeles, M., G. Hayon, and D. Lehmann, Modeling compositionality by dynamic binding of synfire chains, *Journal of Computational Neuroscience*, *17*, 179–201, 10.1023/B:JCNS.0000037682.18051.5f, 2004.

Bi, G., and M. Poo, Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type, *Neural Computation*, *9*, 503–514, 1997.

Bill, J., Self-stabilizing network architectures on a neuromorphic hardware system, Diploma thesis (English), University of Heidelberg, HD-KIP-08-44, 2008.

Brette, R., and W. Gerstner, Adaptive exponential integrate-and-fire model as an effective description of neuronal activity, *J. Neurophysiol.*, *94*, 3637 – 3642, doi:NA, 2005.

Brüderle, D., Neuroscientific modeling with a mixed-signal VLSI hardware system, Ph.D. thesis, 2009.

Brunel, N., Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons, *Journal of Computational Neuroscience*, *8*(3), 183–208, 2000.

Brüderle, D., et al., A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems, *Biological Cybernetics*, *104*, 263–296, 10.1007/s00422-011-0435-9, 2011.

Buxhoeveden, D. P., and M. F. Casanova, The minicolumn hypothesis in neuroscience, *Brain*, *125*(5), 935–951, doi:10.1093/brain/awf110, 2002.

Davison, A. P., D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, PyNN: a common interface for neuronal network simulators, *Front. Neuroinform.*, *2*(11), 2008.

Dayan, P., and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, The MIT press, Cambride, Massachusetts, 2001.

Destexhe, A., Self-sustained asynchronous irregular states and up–down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons, *Journal of Computational Neuroscience*, *27*(3), 493–506, 2009.

Destexhe, A., M. Rudolph, and D. Pare, The high-conductance state of neocortical neurons in vivo, *Nature Reviews Neuroscience*, *4*, 739–751, 2003.

Diesmann, M., M.-O. Gewaltig, and A. Aertsen, Stable propagation of synchronous spiking in cortical neural networks, *Nature*, *402*, 529–533, 1999.

Gerstner, W., and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.

Goedeke, S., and M. Diesmann, The mechanism of synchronization in feed-forward neuronal networks, *New Journal of Physics*, *10*(1), 015,007, 2008.

Grübl, A., VLSI implementation of a spiking neural network, Ph.D. thesis, Ruprecht-Karls-University, Heidelberg, document No. HD-KIP 07-10, 2007.

Hubel, D. H., and T. N. Wiesel, Functional architecture of macaque monkey visual cortex., *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, *198*(1130), 1–59, 1977.

Jeltsch, S., Computing with transient states on a neuromorphic multi-chip environment, Diploma thesis, University of Heidelberg, 2010.

Kremkow, J., L. U. Perrinet, G. S. Masson, and A. Aertsen, Functional consequences of correlated excitatory and inhibitory conductances in cortical networks., *Journal of computational neuroscience*, *28*(3), 579–594, doi:10.1007/s10827-010-0240-9, 2010.

Kumar, A., S. Schrader, A. Aertsen, and S. Rotter, The high-conductance state of cortical networks, *Neural Computation*, *20*(1), 1–43, 2008.

Kumar, A., S. Rotter, and A. Aertsen, Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding., *Nature reviews. Neuroscience*, *11*(9), 615–627, doi:10.1038/nrn2886, 2010.

Lee, A. K., I. D. Manns, B. Sakmann, and M. Brecht, Whole-cell recordings in freely moving rats, *Neuron*, *51*, 399–407, 2006.

Lundqvist, M., M. Rehn, M. Djurfeldt, and A. Lansner, Attractor dynamics in a modular network model, *Network: Computation in Neural Systems*, *17*(3), 253–276, 2006.

Markram, H., Y. Wang, and M. Tsodyks, Differential signaling via the same axon of neocortical pyramidal neurons., *Proceedings of the National Academy of Sciences of the United States of America*, *95*(9), 5323–5328, 1998.

Millner, S., A. Grübl, K. Meier, J. Schemmel, and M.-O. Schwartz, A vlsi implementation of the adaptive exponential integrate-and-fire neuron model, *Advances in Neural Information Processing Systems*, *23*, 1642–1650, 2010.

Mountcastle, V. B., An organizing principle for cerebral function: The unit module and the distributed system, in *Neuroscience, Fourth Study Program*, edited by F. O. Schmitt, pp. 21–42, MIT Press, Cambridge, MA, 1979.

Okun, M., and I. Lampl, Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities, *Nat Neurosci*, *11*(5), 535–7, 2008.

Sanchez-Vives, M. V., and D. A. McCormick, Cellular and network mechanisms of rhythmic recurrent activity in neocortex., *Nature neuroscience*, *3*(10), 1027–1034, doi:10.1038/79848, 2000.

Schemmel, J., D. Brüderle, K. Meier, and B. Ostendorf, Modeling synaptic plasticity within networks of highly accelerated I&F neurons, in *Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS'07)*, IEEE Press, 2007.

Schemmel, J., J. Fieres, and K. Meier, Wafer-scale integration of analog neural networks, in *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, 2008.

Schemmel, J., D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, A wafer-scale neuromorphic hardware system for large-scale neural modeling, *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS"10)*, pp. 1947–1950, 2010.

Shinomoto, S., I. Fujita, Y. Miyazaki, Y. Miyazaki, H. Tamura, and H. Tamura, Regional and laminar differences in in vivo firing patterns of primate cortical neurons, *J Neurophysiol*, *94*, 567–575, 2005.

Timofeev, I., and M. Bazhenov, *Trends in Chronobiology Research*, chap. 1, Nova Science Publishers, Inc., 2005.

Touboul, J., and R. Brette, Dynamics and bifurcations of the adaptive exponential integrate-and-fire model, *Biological Cybernetics*, *99*, 319–334, 2008.

Tsunoda, K., Y. Yamane, M. Nishizaki, and M. Tanifuji, Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns., *Nat Neurosci*, *4*(8), 832–838, doi:10.1038/90547, 2001.

Vogels, T. P., and L. F. Abbott, Signal propagation and logic gating in networks of integrate-and-fire neurons, *J Neurosci*, *25*(46), 10,786–95, 2005.

Vogginger, B., Testing the operation workflow of a neuromorphic hardware system with a functionally accurate model, Diploma thesis, University of Heidelberg, 2010.

Wendt, K., M. Ehrlich, C. Mayr, and R. Schüffny, Abbildung komplexer, pulsierender, neuronaler netzwerke auf spezielle neuronale VLSI hardware, in *DASS'07: Proceedings of Dresdener Arbeitstagung Schaltungs- und Systementwurf*, pp. 127–132 (german), 2007.

Acknowledgments

Prof. Dr. Karlheinz Meier for the opportunity to work in the Electronic Vision(s) group, for his unwavering support and for his unbound commitment to our landmark project and its people.

Dr. Johannes Schemmel for the hardware design and for rigorous explanations

Mihai for intense discussions and general awesomeness

Bernie for providing help and expertise on many occasions

Eric for for technical support and strong opinions

Thomas for general helpfulness.

Sebastian for general helpfulness. Also, climbing

Daniel for thorough advice. Also, foosball

Björn for flawless administration and organization

The hardies for general helpfulness. Also, the hardware

The softies for the supplies. Also, the software

Takkara for his support

Arne for cookies. Also, Sweden

Sandra for tea. Also, Sweden

My family.

Statement of Originality (Erklärung):

I certify that this thesis, and the research to which it refers, are the product of my own work. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Ich versichere, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, June 15, 2011

.....
(signature)