Multi-timescale Synaptic Plasticity on Mixed-Signal Neuromorphic Silicon

Thesis

submitted in partial fulfillment of the requirements for the degree
Master of Science

Graduate Training Center of Neuroscience
Faculty of Science
Faculty of Medicine

Presented by
Amani Atoui
from Kafarkela, Lebanon

Tuebingen, September 30th, 2024

Thesis Advisor: Dr. Johannes Schemmel

Electronic Vision Group
Kirchhoff's Institute for Physics
University of Heidelberg

Second Reader: Dr. Martin Giese

Hertie Institute for Clinical Brain Research
University of Tübingen

Disclosures:

- I affirm that I have written the thesis myself and have not used any sources and aids other than those indicated.
- I affirm that I have not included data generated in one of my laboratory rotations and already presented in the respective laboratory report.

Signature ____

Abstract

One application of neuromorphic computers is providing an energy-efficient solution for computer simulations of neural networks in the field of computational neuroscience. BrainScaleS-2 is a mixed-signal neuromorphic platform that promises accelerated and faithful emulation of biological neural networks. In this work, we aim to emulate the synaptic tagging and capture plasticity rule for a single synapse on BrainScaleS-2. The three main steps for emulating this rule are emulating the neural dynamics, emulating the calcium dynamics, and implementing the equations of the plasticity rule. The first two steps are achieved using the neuron and the adaptation circuits respectively of the analog core of BrainScaleS-2 that emulates the neuron dynamics based on the adaptive exponential leaky integrate-and-fire model. The third step relies on differential equations solved numerically on the digital plasticity processor of BrainScaleS-2. The main hardware constraints are the update timestep used to solve the differential equations and the use of finite arithmetic. Using a higher update timestep and stochastic rounding, we show that BrainScaleS-2 can faithfully emulate a single synapse that follows the synaptic tagging and capture plasticity rule for four stimulation protocols. The results show that almost no statistically significant results exist between the simulation and the emulation schemes, and that the variability of the test statistics mostly stems from the spikes. We conclude our work by reaffirming the role of BrainScaleS-2 in speeding-up the emulation of neural dynamics in computational neuroscience.

Contents

List of Figures

List of Tables

Acronyms

AdEx adaptive exponential leaky integrate-and-fire.

AI artificial intelligence.

CADC column-parallel analog-to-digital converter.

CMOS complementary metal oxide semiconductor.

DAC digital-to-analog converter.

FPGA field-programmable gate array.

ISI interspike interval.

LIF leaky integrate-and-fire.

LSB least significant bits.

LTD long-term depression.

LTM long-term memory.

LTP long-term potentiation.

MADC membrane analog-to-digital converter.

PPU plasticity processing unit.

PRP plasticity-related proteins.

RN round-to-nearest.

SIMD single instruction, multiple data.

SLFS strong low-frequency stimulation.

SNN spiking neural network.

SR stochastic rounding.

STC synaptic tagging and capture.

STDP spike-timing dependent plasticity.

STET strong tetanic stimulation.

SU stochastic update.

WLFS weak low-frequency stimulation.

WTET weak tetanic stimulation.

Motivation

Neuromorphic computing is a computing technology inspired by the human brain's architecture. Many reasons have encouraged the computing community to develop such technologies (Schuman et al., 2022). On one hand, the von Neumann computer architecture possesses many limitations in terms of parallel operations and memory/processing separation. These limitations hinder the computing performance and efficiency in fields related to computational neuroscience, artificial intelligence (AI), and robotics (Indiveri and Liu, 2015). On the other hand, the human brain is known for its high computing abilities with low power consumption. This superior performance of the brain can be attributed to many reasons of which we know the memory-processing collocation that allows parallel computations and the event-driven operation of networks enabled by sparse spikes (Schuman et al., 2022).

There exist fundamental operational differences for neuromorphic computers against standard von Neumann architectures starting with the parallel operations in neuromorphic computers where all neurons can be operated simultaneously, and the collocated memory/processing (Schuman et al., 2022). Additionally, neuromorphic computers are expected to achieve inherent scalability by connecting multiple chips as a single large computer. One of the most attractive features in neuromorphic computing is the sparse activity of spikes between neurons that enables event-driven computation. This feature is believed to achieve power and energy efficiency as only a small portion of the system would be active at a given time. In their general architecture, neuromorphic computers implement the so-called spiking neural network (SNN)s, in which the functionality of neurons and synapses is biologically plausible. These networks account for timing in their operation where information propagates asynchronously due to the delay differences between neurons and synapses.

Many physical realizations of neuromorphic computers have been developed in the past years. These systems are designed based on a variety of choices depending on the end goals and fields that these platforms are targeted at. BrainScaleS-2 (Pehle et al., 2022) is a neuromorphic platform primarily targeted at faithfully emulating biologically plausible neural dynamics for the advancement of AI, robotics, and computational neuroscience. Positive consequences follow from the advancement of computational neuroscience, ranging from understanding the human nervous system through simulating large neural networks to complementing medical applications. Computer simulations are an indispensable part of computational neuroscience, specifically for understanding neuron circuits through synaptic plasticity (Zenke and Gerstner, 2014). The challenge lies in simulating large networks with an adequate simulation timestep, which requires long experiments with large waiting times to capture the timescale of long-term plasticity (Zenke and Gerstner, 2014). Here comes the role of specialized neuromorphic hardware, such as BrainScaleS-2, that can emulate neural dynamics at accelerated timescales compared to real time.

Another useful consequence of the advancement of theoretical neuroscience is deriving learning rules for machine intelligence. Since the neuromorphic hardware architecture is inspired by the human brain, we argue that neuromorphic computers must be connected to special neuromorphic algorithms that can effectively exploit their characteristics. Our focus is on biologically-plausible plasticity algorithms that would eventually be used as learning rules for SNNs to perform a variety of tasks in the fields of machine learning and robotics. One of the well-established rules is spike-timing dependent plasticity (STDP) which adjusts weights based on the relative spike timings of presynaptic and postsynaptic neurons. The STDP learning rule has proven effective as a clustering mechanism, specifically as a spike sorter in brain machine interface applications (Schuman et al., 2022). However, experiments show that the temporal order is only relevant in a small regime to presynaptic activation, and the synaptic modification seem to be independent of the postsynaptic spiking activity (Tetzlaff et al., 2012). An alternative to STDP is calcium-based plasticity formulated as the synaptic tagging and capture (STC) hypothesis (Tetzlaff et al., 2012).

The strength of the STC plasticity scheme lies in its dependence on different synaptic states and timescales (Redondo and Morris, 2011). Specifically, it asserts that long-lasting synaptic changes do not have to be triggered at close time instants, and these changes could be tied to events in the past, present, and future. The STC plasticity mechanism has proven its plausibility in many behavioral, and consequently, realistic settings. For example, it has been linked with the improvement of memory recall in recurrent neural networks as well as the enhancement of learning after a time break from memory encoding (Luboeinski and Tetzlaff, 2021). Behavioral tagging has also been identified as the analog of synaptic tagging in the STC hypothesis in many behavioral experiments (Okuda et al., 2021) such as the persistence of long-term potentiation (LTP) and the encoding of everyday spatial memory in rats (Wang et al., 2010).

Motivated by the previous implementations of plasticity rules on BrainScaleS-2 and the variety of neuron dynamics that the corresponding neuron circuits can emulate, we aim to implement the STC plasticity rule on BrainScaleS-2. In this work, we focus on emulating the rule for a single synapse in attempt to emulate at later stages neural networks that can exploit the STC rule. In the next sections, we describe the methods followed to implement the rule on a single synapse, and we present the results that seem to faithfully emulate the rule's dynamics. We discuss the constraints of our algorithm and present future extensions to our work including emulating a full recurrent network that would provide reliable conclusions on memory consolidation as presented in (Luboeinski and Tetzlaff, 2021) or even performing real-world tasks to achieve machine intelligence. We conclude our work by re-affirming the flexibility of BrainScaleS-2, and the role it serves in computational neuroscience and the advancement of bio-inspired computing.

Background

In this section, we introduce the relevant concepts that are necessary for emulating the STC plasticity rule of neurons on BrainScaleS-2. It is significant to note that our goal in this work is highly interdisciplinary. This is clear from the topics that we covered in the motivation section and that would be described in more details in this section. Nevertheless, these topics will converge smoothly throughout this work.

2.1 Memory and Synaptic Tagging and Capture Hypothesis

An engram, or a memory trace, is the physical entity in the brain that stores information over time and enables memories to be expressed (Redondo and Morris, 2011). The physical substrate of engrams for the persistence of long-term memory (LTM) is a change in the biophysical and structural efficacy of synapses within neural networks. The persistence of these engrams depends on many determinants, but the immediate determinant is a consolidation process that gets activated after memory encoding and initial storage. This consolidation process requires specific conditions to be activated and enables memory traces to be stabilized. The consolidation process has two components; at the high level between cortices and neural circuits, systems consolidation involves dynamic interactions between the neocortical and hippocampal circuits for creating or updating engrams. Synaptic or initial consolidation, on the other hand, is performed at the cellular level, and it is concerned with synaptic strength for duration ranging from minutes to hours. The dual framework of memory consolidation at the cellular and systems levels is considered crucial for the persistence of memory (Redondo and Morris, 2011). For example, the activation of excitatory glutamatergic synapses in the hippocampus triggers a variety of cellular cascades which consequently play an important role in learning and memory.

A theory that explains a significant part of synaptic consolidation is the STC hypothesis. It was first suggested by Frey and Morris (1997) and later revised by Redondo and Morris (2011) to account for various limitations and challenges to the initial hypothesis. In the theory on synaptic consolidation, LTP and long-term depression (LTD) involve two phases (Lamprecht and LeDoux, 2004). The early phase is characterized by the increase in calcium concentration, while the late phase is characterized by insertion of neurotransmitter receptors in the case of LTP and their removal in the case of LTD. In between these two phases, the biophysical and structural alterations of the synapses contribute to stabilizing the levels of synaptic strength. The stabilization depends on the recent history and near future of the neural activity, and both can be dependent or independent of the neural activity occurring during encoding. This extension of the time window that determines memory consolidation is accompanied with the synthesis and distribution of plasticity-related proteins (PRP) induced by other neuronal activities. In the STC hypothesis, the proteins are captured by synaptic tags which are local molecular changes at synapses found to mark these synapses as having experienced synaptic plasticity. Computationally, the STC hypothesis accounts for the several 'states' of a synapse that can reflect the level of synaptic strength and the potential for LTM.

A core concept of the STC hypothesis is that memory encoding creates a potential for LTM but not a commitment itself (Redondo and Morris, 2011; Luboeinski and Tetzlaff, 2021). In behavioural and experimental neuroscience, it is intuitive to think about memory formation as a sequence of events triggered at a moment in time. For example in some experiments, a strong firing in specific neurons could be attributed to animals learning by receiving rewards, and this would immediately lead to consolidation between the different events at the experiment time. However, this form of learning is somehow restricted to laboratory experiments. Realistic events occur at different times such that their memorability can be affected by other past or future events. If memory consolidation were always triggered by all these events and stimuli, the memory systems of the brain would be simultaneously handling numerous consolidation cascades, which is not feasible for neurons as the capacity of the nervous system would be reached immediately (Redondo and Morris, 2011; Tetzlaff et al., 2012). The STC hypothesis accounts for this timing issue; it asserts that memory consolidation involves a set of mechanisms that can but do not have to be triggered at a single moment in time. One of the supporting experiments for this claim was that LTP was induced even during the inhibition of protein synthesis (Frey and Morris, 1997).

According to the STC hypothesis, LTP occurs over the following steps after neurons have fired sufficiently. First, early LTP is expressed, and the synapse is marked with a local synaptic tag. Second, the PRP are synthesized and distributed. Third, these proteins are captured by the tagged synapses which allow the stabilization of the synaptic strength in the late LTP phase. The time course of synaptic consolidation is also influenced by the neural network. More specifically, if there was a prior activity of the neuron that has already upregulated the availability of PRPs at a synapse, these proteins will be captured by local synaptic tags and ensure the stabilization of the synaptic component of a new memory trace without the need to produce proteins for the new activity. This suggests that neural activities from weak stimulations can be consolidated if there is a prior presence of proteins. Conversely, if neural activity that induces PRP synthesis occurs later after early LTP and tag setting, stabilization of the synaptic strength will occur at this later time with the temporal duration of the tag being the main determinant of the persistence of LTM. This also suggests that role of the networks in consolidating weak and strong neural activities that occur within a short time window.

2.2 Neuron Models

2.2.1 Leaky Integrate-and-Fire Model

Integrate-and-fire models (Gerstner and Kistler, 2002) are single-compartment neuron models which follow the assumption that information in neurons is contained in the occurrence spikes. The shape of the action potential is almost the same among all neurons and does not carry information. Instead, action potentials are considered as events. The simplest integrate-and-fire neuron model uses a linear differential equation to describe the dynamics of the membrane potential V. Equation 2.1 uses conservation of the currents across the neuron membrane between an external injected current I that models synaptic input, the leak current controlled by the membrane resistance R, and the capacitive current formed by the membrane insulation and charges giving rise to a membrane time constant τ_{mem}:

$$\tau_{\mathrm{mem}} \frac{dV(t)}{dt} = V_{\mathrm{rev}} - V(t) + \mathrm{R} \cdot \mathrm{I}(t) \tag{2.1}$$

with V_{rev} being the reversal potential. Besides the differential equation, the second ingredient for the integrate-and-fire model is a spiking mechanism defined by a threshold ϑ for firing an action potential:

$$V(t^{(f)}) = \vartheta, \tag{2.2}$$

after which the membrane potential resets to a fixed value V_r:

$$V(t^{(f)+}) = V_r \tag{2.3}$$

The main assumption of this model about the shape of the action potential and its leak term coin its name, the leaky integrate-and-fire (LIF) model. It is a highly simplified model that is accurate at generating spikes, but it also neglects many features observed in biological neurons. If refractoriness is accounted for in the LIF model through a refractory period t_{ref}, the performance of the model in spike generation can be further improved (Gerstner and Kistler, 2002) as in the case of the neuron model by Luboeinski and Tetzlaff (2021).

2.2.2 Adaptive Exponential Leaky Integrate-and-Fire Model

The adaptive exponential leaky integrate-and-fire (AdEx) model proposed by Brette and Gerstner (2005) is a spiking neuron model with two variables, the membrane potential V and the adaptation current I_{adapt}:

$$\tau_{\mathrm{mem}} \frac{dV(t)}{dt} = V_{\mathrm{rev}} - V(t) + \Delta_T \exp\left(\frac{V(t) - V_T}{\Delta_T}\right) - \mathrm{R} \cdot \mathrm{I}_{adapt}(t) + \mathrm{R} \cdot \mathrm{I}(t) \tag{2.4}$$

$$\tau_{adapt} \frac{dI_{adapt}(t)}{dt} = a \cdot (V(t) - V_{\mathrm{rev}}) - I_{adapt}(t) \tag{2.5}$$

where τ_{mem} is the membrane time constant, V_{rev} is the reversal potential, Δ_T is the slope factor, V_T is the threshold voltage, R is the membrane resistance, I is the input current, τ_{adapt} is the time constant of the adaptation current, and a is the subthreshold adaptation term.

In this model, the membrane potential is coupled with the adaptation current to give rise to realistic neuron dynamics. More specifically, the exponential nonlinearity in eq. (2.4) models the process of spike generation and upswing of the action potential. The downswing of the action potential is modelled by discrete resets similar to the LIF model. That is, the membrane potential integrates until a threshold ϑ is reached, and a spike is elicited at $t = t_f$. At this instant, the membrane potential is reset to a fixed voltage V_r, and the adaptation current is incremented by a fixed value b.

$$V(t = t^{(f)+}) = V_r \tag{2.6}$$

$$I_{adapt}(t = t^{(f)+}) = I_{adapt} + b \tag{2.7}$$

where b is the spike-triggered adaptation term. These hard resets in eq. (2.6) and eq. (2.7) along with the differential equations in eq. (2.4) and eq. (2.5) give rise to rich dynamics including adapting, bursting, delayed spike initiation, initial bursting, fast spiking, and regular spiking (Brette and Gerstner, 2005).

2.3 Neuromorphic Hardware

The current general-purpose computers are designed to excel in a setting where the algorithms can be specified absolutely and the symbols of the computation can be assigned unambiguously (Douglas et al., 1995). For example in biological neural networks, the systems have to be translated to a mathematical form including the state variables and parameters. Time should also be abstracted, and natural phenomena such as noise must be specified. These computers use the digital representation which generally means that they restore information in the form of bits. The advantages of computing in digital systems are the precise computation and the ease of storage and display of results (Douglas et al., 1995).

However, the real world is less rigid, and the actual behavior of phenomena emerges from a complex network of factors (Douglas et al., 1995). This possibly means that real-world neural computations cannot be achieved using general-purpose digital methods with the symbolic approach. The early vision of neuromorphic hardware was to emulate the function and organization of biological neural networks using analog electronic circuits fabricated using the complementary metal oxide semiconductor (CMOS) technology. Unlike digital systems, analog circuits directly rely on the physics of the CMOS circuits to obtain physical processes that behave like neural processes. The advantage of analog computing in this case lies in its efficiency; instead of using many components for storage and digital clock cycles to execute algorithms on digital computers, these would naturally arise out of the physical processes in analog circuits (Douglas et al., 1995).

2.3.1 BrainScaleS-2

BrainScaleS-2 is a mixed-signal neuromorphic computing platform, which means that it uses both analog and digital circuits in its hardware architecture. The architecture features an analog neuromorphic core that emulates neural dynamics at 1000-fold accelerated timescales compared to real time, thus promising accelerated emulations of neural networks. Alongside is a digital circuit that ensures flexible control, calculations, and plasticity (Pehle et al., 2022). BrainScaleS-2 encompasses 512 neuron circuits that can emulate a variety of dynamics observed in biological neurons (Billaudelle et al., 2022). The physical design of the chip mainly consists of 4 quadrants, and each quadrant encloses 128 silicon neurons. Neurons can form synaptic connections through a synaptic crossbar with 256 rows and 128 columns. In other words, each neuron can communicate by default with 256 different neurons in a network.

In the analog core of BrainScaleS-2, the neuron circuits are designed based on the AdEx model (Brette and Gerstner, 2005). Due to the flexibility of BrainScaleS-2 neuron circuit design and variety of circuit parameters, the neuron model can also be simplified to the LIF model. Each neuron circuit can be configured individually through 24 analog parameters using an on-chip digital-to-analog converter digital-to-analog converter (DAC) with 10-bit resolution (Pehle et al., 2022). These analog parameters control the potentials and conductances provided in equations 2.4 and 2.5. The general unit of measurement for all the analog states and parameters is the least significant bits (LSB), which is the smallest weighted bit in the number that represents that analog state or parameter. Eventually, these numbers map to a voltage range but requires tuning.

On the other hand, BrainScaleS-2 features two digital control and plasticity processors, each referred to as plasticity processing unit (PPU), to ensure a flexible digital-control architecture. The PPU can be used to implement programmable plasticity rules, automatic on-chip calibration, parallel readout of analog observables for learning, and a variety of other use cases (Pehle et al., 2022). For the implementation of the plasticity rules, these processors can read and write the digital state of the neurons and synapses by means of single instruction, multiple data (SIMD) vector extensions. The SIMD units can perform fixed-point and integer arithmetic operations on vectors of either 128 8-bit entries or 64 16-bit entries. For carrying out any experiment on BrainScaleS-2, a field-programmable gate array (FPGA) is used for real-time control, specifically buffering of external stimuli such as spikes and output data (Pehle et al., 2022).

The analog traces of the neuron dynamics can be sampled by means of a membrane analog-to-digital converter (MADC) and a column-parallel analog-to-digital converter (CADC). The MADC has a higher sampling resolution compared to the CADC, which makes the MADC more suitable in initial experiments for tuning the neuron parameters. However, the CADC provides a faster alternative that can be used in plasticity rules run on the PPU.

BrainScaleS-2 supports correlation-based plasticity rules such as STDP through its built-in analog sensor circuits within each synapse (Friedmann et al., 2016). Additionally, the CADC together with the PPU facilitate the implementation of different plasticity rules on BrainScaleS-2. For example, several versions of STDP-based learning rules were implemented (Wunderlich et al., 2019), as well as learning rules that include structural plasticity with a form of stochasticity (Billaudelle et al., 2021). At a higher level of learning, BrainScaleS-2 has been proven useful in carrying out optimizations for learning to learn scenarios for spiking neural networks through exploiting the accelerated timescales compared to biological time (Bohnstingl et al., 2019).

2.4 Stochastic Rounding

Rounding is a function that maps a given number to another number while keeping the resulting number close to its original value. Rounding is needed to make numbers simpler and more usable, but also to represent numbers in finite precision number systems. The most common rounding function is the round-to-nearest (RN) mode, which as its name suggests, rounds the given number to a nearest value that can be represented by the system at hand (Croci et al., 2022). The RN rule is deterministic, meaning that the output of the rounding function depends only on the given number to be rounded, and repeating the rounding yields the same result (Croci et al., 2022).

In attempt to reduce the accumulation of round-off errors produced by the RN mode, another rounding mode referred to as stochastic rounding (SR) was first proposed in an abstract in 1949 (Croci et al., 2022; Forsythe, 1950). In the SR mode, a given number is mapped to either of the two nearest representable numbers with a probability that depends on the distances of the given number to the nearest numbers. Particularly, let F be a subset of $\mathbb{R}$ and fl be a rounding operator that maps a real number $x \in \mathbb{R}$ to one of the two nearest numbers in F. The nearest numbers can be defined as (Croci et al., 2022):

$$\lfloor x \rfloor = \max\{y \in F : y \leq x\} \tag{2.8}$$

$$\lceil x \rceil = \min\{y \in F : y \geq x\} \tag{2.9}$$

If $x \in F$, then $\lfloor x \rfloor = \lceil x \rceil = x$, and if $x \notin F$, then the 2 numbers $\lfloor x \rfloor$ and $\lceil x \rceil$ are adjacent in F with a distance of ϵ_M referred to as the machine epsilon. In the latter case, SR is defined as (Croci et al., 2022):

$$\mathrm{fl}(x) = \begin{cases} \lceil x \rceil \text{ with probability } q(x) \\ \lfloor x \rfloor \text{ with probability } 1 - q(x) \end{cases} \tag{2.10}$$

where $q(x)$ is set to:

$$q(x) = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor} = \frac{x - \lfloor x \rfloor}{\epsilon_M} \tag{2.11}$$

This form of SR is an interesting rounding mode from a numerical point of view. Particularly, SR ensures zero-mean rounding errors and produces smaller errors compared to RN especially in situations where RN produces rounding errors of one sign (Croci et al., 2022). More importantly, SR is immune to stagnation, a phenomenon where a sequence of small updates relative to large quantities is lost as these updates cannot be represented by the computing system. To demonstrate stagnation, suppose we want to compute $1 + 0.1$ in 1-digit base-10 arithmetic (Croci et al., 2022). With RN, $\mathrm{fl}(1 + 0.1) = 1$; this result is fixed and the error is always 0.1. Using SR, we define $\mathrm{fl}(1 + 0.1) = 1$ with a probability of 0.9, and $\mathrm{fl}(1 + 0.1) = 2$ with a probability of 0.1 such that the expected result is $1 * 0.9 + 2 * 0.1 = 1.1$. In other words, using this rounding mode reduces on average the accumulated error, especially if there are other computations that depend on this rounding.

Despite being proposed earlier, the interest in SR is currently expanding as it proved useful mainly in applications related to machine learning and solving ordinary differential equations on finite precision systems. For example, Hopkins et al. (2020) demonstrated the importance of rounding in solving ordinary differential equations in neuroscience by using SR to produce accurate spike timings in the Izhikevich neuron model. Other useful domains of application of SR include numerical linear algebra, partial differential equations, quantum mechanics, and quantum computing (Croci et al., 2022). A primary consideration worth mentioning is the energy-efficiency in all areas of computing, which caused the shift towards reduced precision arithmetic and the need to overcome the resulting drawbacks using SR.

When considering replacing RN with SR, it is crucial to consider the properties of RN that are not necessarily preserved in SR, and whether the computations of interest depend on these properties. There are several properties listed in (Croci et al., 2022), but we only focus on the properties that would be of concern in our application:

1. For SR, if $x \in F$, then $\mathrm{fl}(|x|) \neq |\mathrm{fl}(x)|$ and $\mathrm{fl}(-x) \neq -\mathrm{fl}(x)$.
2. SR is not monotonic, meaning that $x \leq y$ does not necessarily imply that $\mathrm{fl}(x) \leq \mathrm{fl}(y)$.

Methods

In this section, we describe the methods for emulating a single synapse that obeys the STC plasticity rule on BrainScaleS-2. We first introduce the theoretical model that we rely on from Luboeinski and Tetzlaff (2021) and illustrate how this model can be configured on the circuits of BrainScaleS-2. We then describe in details the hardware emulation of the neuron dynamics, calcium dynamics, and STC plasticity rule.

3.1 Model and Experiment Setup

3.1.1 Single-Synapse Model

In this section, we describe the biological mechanisms governing the used synaptic model depicted in figure 3.1 which incorporates the calcium dynamics according to the STC hypothesis (Luboeinski and Tetzlaff, 2021). In this model, a presynaptic neuron j is connected via a synapse to a postsynaptic neuron i. Spikes arriving from neuron j at times t_j^n along with the initial total synaptic weight w_{ji} induce a change in the postsynaptic membrane potential V_i. If the membrane potential exceeds a threshold, a postsynaptic spike is elicited at time t_i^m. The presynaptic and postsynaptic spikes drive the postsynaptic calcium concentration c_{ji} which induces early-phase plasticity, quantified by the early-phase weight h_{ji}. In case the synapse is tagged, and the proteins synthesized p_i during the early phase are sufficient, late-phase plasticity is induced, quantified by the late-phase weight z_{ji}. Consequently, the total synaptic weight w_{ji} is represented by a weighted sum of the early-phase weight h_{ji} and the late-phase weight z_{ji}.

All equations governing the single-synapse plasticity are adopted from Luboeinski and Tetzlaff (2021). The dynamics of the postsynaptic membrane potential follow the LIF neuron model:

$$\tau_{\mathrm{mem}} \frac{dV_i(t)}{dt} = V_{\mathrm{rev}} - V_i(t) + \mathrm{R} \cdot (\mathrm{I_{bg}(t)} + \mathrm{I_{stim}(t)} + \mathrm{I_{syn,i}(t)}) \tag{3.1}$$

with reversal potential V_{rev}, membrane time constant τ_{mem}, membrane resistance R, external background current $\mathrm{I_{bg}(t)}$, external stimulus current $\mathrm{I_{stim}(t)}$, and synaptic current $\mathrm{I_{syn,i}(t)}$. The synaptic current $\mathrm{I_{syn}}$ is defined as:

$$\mathrm{I_{syn,i}(t)} = \sum_j \sum_{t_j^k} w_{ji} \cdot \exp\left(-(\mathrm{t} - \mathrm{t_j^k} - \mathrm{t_{ax,delay}})/\tau_{\mathrm{syn}}\right) \tag{3.2}$$

with axonal time delay $\mathrm{t_{ax,delay}}$ and synaptic time constant τ_{syn}. For a single synapse, the background current and the stimulation current are set to zero. The membrane

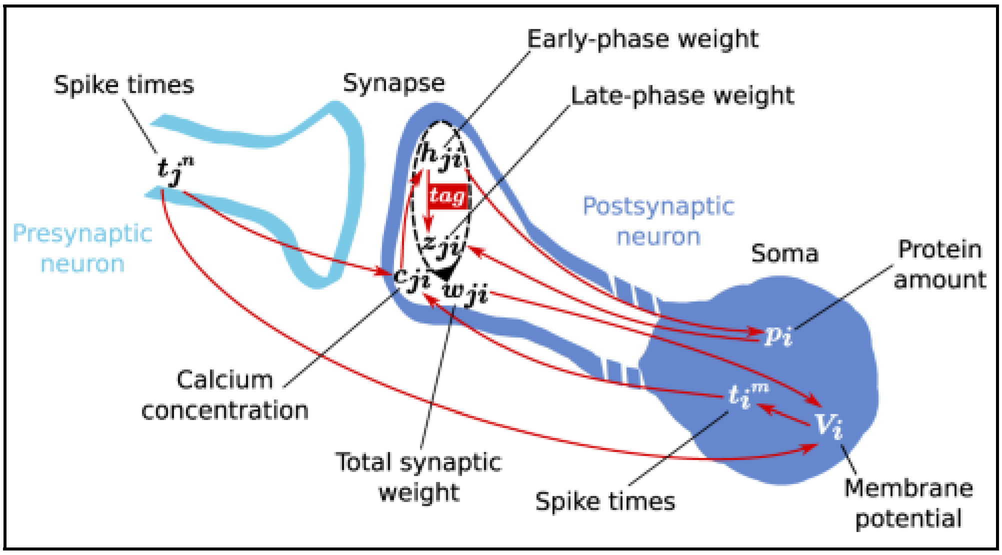


Figure 3.1: The synaptic model integrating various mechanisms of calcium-dependent synaptic plasticity and the STC hypothesis. Modified from Luboeinski and Tetzlaff (2021).

potential for the single synapse thus reduces to eq. (3.3).

$$\tau_{\text{mem}} \frac{dV_i(t)}{dt} = V_{\text{rev}} - V_i(t) + \mathrm{R} \cdot \mathrm{I}_{\mathrm{syn,i}}(\mathrm{t}) \tag{3.3}$$

In the presence of synaptic input, if the membrane potential V exceeds a threshold potential V_{th}, a spike is elicited, and V is set to a reset potential V_{reset} for a refractory period t_{ref}.

The calcium dynamics follow eq. (3.4) with the calcium time constant τ_{c}, the contribution of presynaptic spikes c_{pre}, the contribution of postsynaptic spikes c_{post}, and the delay of calcium concentration triggered by presynaptic spikes $t_{\text{c,delay}}$.

$$\frac{dc_{ji}(t)}{dt} = -\frac{c_{ji}(t)}{\tau_{\text{c}}} + c_{\text{pre}} \sum_n \delta(t - t_j^n - t_{\text{c,delay}}) + c_{\text{post}} \sum_m \delta(t - t_i^m), \tag{3.4}$$

The dynamics of the early-phase weight are governed by eq. (3.5) with $\Theta[.]$ being the Heaviside function and τ_h being a time constant. The first term of eq. (3.5) describes a relaxation of the early-phase weight to its initial value h_0, the second term describes early-phase LTP with rate γ_{p} for calcium concentration above the potentiation threshold θ_{p}, and the third term describes early-phase LTD with rate γ_{d} for calcium concentration above the depression threshold θ_{d}.

$$\begin{aligned} \tau_h \frac{dh_{ji}(t)}{dt} = {} & 0.1\ (h_0 - h_{ji}(t)) + \gamma_{\text{p}}(1\ \text{nC} - h_{ji}(t)) \cdot \Theta[c_{ji}(t) - \theta_{\text{p}}] \\ & - \gamma_{\text{d}} h_{ji}(t) \cdot \Theta[c_{ji}(t) - \theta_{\text{d}}] + \xi(t), \end{aligned} \tag{3.5}$$

The term $\xi(t) = \sqrt{\tau_h \cdot (\Theta[c_{ji}(t) - \theta_{\text{p}}] + \Theta[c_{ji}(t) - \theta_{\text{d}}])}\, \sigma_{\text{pl}}\, \Gamma(t)$ describes the calcium-dependent noise-driven fluctuations with standard deviation σ_{pl}, and Gaussian white noise $\Gamma(t)$ with mean zero and variance $\frac{1}{\Delta t}$, where Δt is the time step for numerical computations. Knowing the early-phase weight, the protein amount is updated using eq. (3.6) where α is the protein synthesis rate and θ_{pro} is the protein synthesis threshold.

$$\tau_p \frac{dp_i(t)}{dt} = -p_i(t) + \alpha\ \Theta\left[\left(\sum_j |h_{ji}(t) - h_0|\right) - \theta_{\text{pro}}\right] \tag{3.6}$$

The dynamics of the late-phase weight depend on the protein amount, early-phase weight, and a tagging threshold θ_{tag}.

$$\begin{aligned}\tau_z \frac{dz_{ji}(t)}{dt} &= p_i(t) \cdot (1 - z_{ji}(t)) \cdot \Theta[(h_{ji}(t) - h_0) - \theta_{\text{tag}}] \\ &\quad - p_i(t) \cdot (z_{ji} + 0.5) \cdot \Theta[(h_0 - h_{ji}(t)) - \theta_{\text{tag}}]\end{aligned} \tag{3.7}$$

Finally, the total synaptic weight is given by:

$$w_{ji}(t) = h_{ji}(t) + h_0 \cdot z_{ji}(t) \tag{3.8}$$

For the rest of the report, we drop all the indices of the model variables, j and i, corresponding to the presynaptic and postsynaptic neurons respectively, since we only deal with a single synapse in this work. The parameters of the model are listed extensively in table 3.1.

3.1.2 Linking STC Variables to BrainScaleS-2

Emulation versus Simulation

In computational neuroscience, computer simulations are usually carried out for deriving and experimenting with neuron models, synapse models, and neural networks. In a simulation, the dynamics of the physical system under consideration is mimicked using another physical system that originally has different dynamics from the physical system under study. More specifically, in a simulation of the STC model, the differential equations that govern the neuron and synapse dynamics of interest are numerically computed using a computer that operates according to an architecture that does not resemble the neuron's architecture, for example a standard von Neumann architecture. In an emulation, on the other hand, a physical system that resembles the one under study is used to mimic the latter system, thus achieving efficient computing. Here, we emulate a single synapse that follows the STC model described in section 3.1.1 on BrainScaleS-2 whose silicon neurons follow the AdEx model.

STC Variables on BrainScaleS-2

The analog core of BrainScaleS-2, primarily consisting of neurons and synapses, allows the emulation of the neuron dynamics, namely the membrane potential V and

Symbol	Value	Description
Δt	0.2 ms	Timestep of numerical integration
t_{end}	8 hours	Total simulation duration
τ_{mem}	10 ms	Membrane time constant
τ_{syn}	5 ms	Synaptic time constant
$t_{ax,delay}$	3 ms	Axonal spike delay
t_{ref}	2 ms	Refractory period
R	10 MΩ	Membrane resistance
V_{rev}	-65 mV	Reversal potential
V_{reset}	-70 mV	Reset potential
V_{th}	-55 mV	Threshold potential for spiking
h_0	0.420 075 nC	Initial early-phase weight
$t_{c,delay}$	0.0188 s	Delay of postsynaptic calcium influx after presynaptic spike
c_{pre}	1	Presynaptic calcium contribution
c_{post}	0.2758	Postsynaptic calcium contribution
τ_c	0.0488 s	Calcium time constant
τ_p	60 min	Protein time constant
τ_z	60 min	Late-phase time constant
γ_p	1645.6	Potentiation rate
γ_d	313.1	Depression rate
θ_p	3	Calcium threshold for potentiation
θ_d	1.2	Calcium threshold for depression
σ_{pl}	0.290 436 $\mathrm{nC\,s^{-1/2}}$	Standard deviation for plasticity fluctuations
α	1	Protein synthesis rate
θ_{pro}	0.210 037 nC	Protein synthesis threshold
θ_{tag}	0.084 014 9 nC	Tagging threshold

Table 3.1: Neuron and synapse model parameters of the simulation scheme. Adapted from Luboeinski and Tetzlaff (2021).

the synaptic current I_{syn}, with spikes injected as digital events. The calcium dynamics c are directly dependent on the presynaptic and postsynaptic spikes with a low time constant in the order of milliseconds that allows the calcium concentration to decay quickly. Therefore, the calcium dynamics should be linked to the analog core of BrainScaleS-2 where spikes are injected. To do that, we exploit the adaptation circuit that was incorporated into BrainScaleS-2 to apply the AdEx model. Specifically, we use the adaptation current whose dynamics greatly resemble those of calcium to emulate the calcium dynamics. The rest of the STC variables have to be computed using the PPU since they have high time constants in the order of seconds and hours, and their dynamics are not directly tied to the neuron dynamics. These variables are explicitly the early-phase weight h, protein amount p, late-phase weight z, and total synaptic weight w. Since these variables rely on the calcium concentration, the latter should be sampled by the CADC at a predefined sampling period δt to be used for weight updates. Finally, the total synaptic weight w is mapped to the hardware weight of the synaptic projection to proceed the experiment with new synaptic weights.

3.1.3 Experiment Configuration

Ideally, a single synapse is formed by two neurons, a presynaptic neuron and a postsynaptic neuron. However, for emulating a single synapse on BrainScaleS-2, one silicon neuron acting as a postsynaptic neuron is configured rather than two. This is because the membrane potential and the synaptic current of only the postsynaptic neuron will be affected by the presynaptic spikes. Therefore, a spike source is configured that plays the role of the presynaptic neuron which would elicit spikes to a configured postsynaptic neuron. However, for the STC rule, additional requirements apply. The adaptation trace generated by the postsynaptic neuron depends on the postsynaptic spikes only, but the calcium trace depends on the presynaptic and postsynaptic spikes. To resolve this, we exploit one of the supported modes by the neurons of BrainScaleS-2 referred to as the “bypass mode”. In this mode, presynaptic spikes are directly translated to postsynaptic spikes with a negligible time delay. We use this mode in another configured neuron connected to the same spike source, referred to as the “parrot neuron”, which ensures that each presynaptic spike induces one and only one postsynaptic spike at almost the same instant. In this way, we mirror the presynaptic spikes and produce an adaptation trace from the postsynaptic spikes of the parrot neuron. The required calcium trace is thus the weighted sum of the two adaptation traces. Calcium is then sampled regularly from the adaptation trace to update the early-phase weight, protein amount, late-phase weight, and total synaptic weight. Finally, the computed total synaptic weight is mapped to the hardware weight that is used to define the projection of the presynaptic neuron to the postsynaptic neuron.

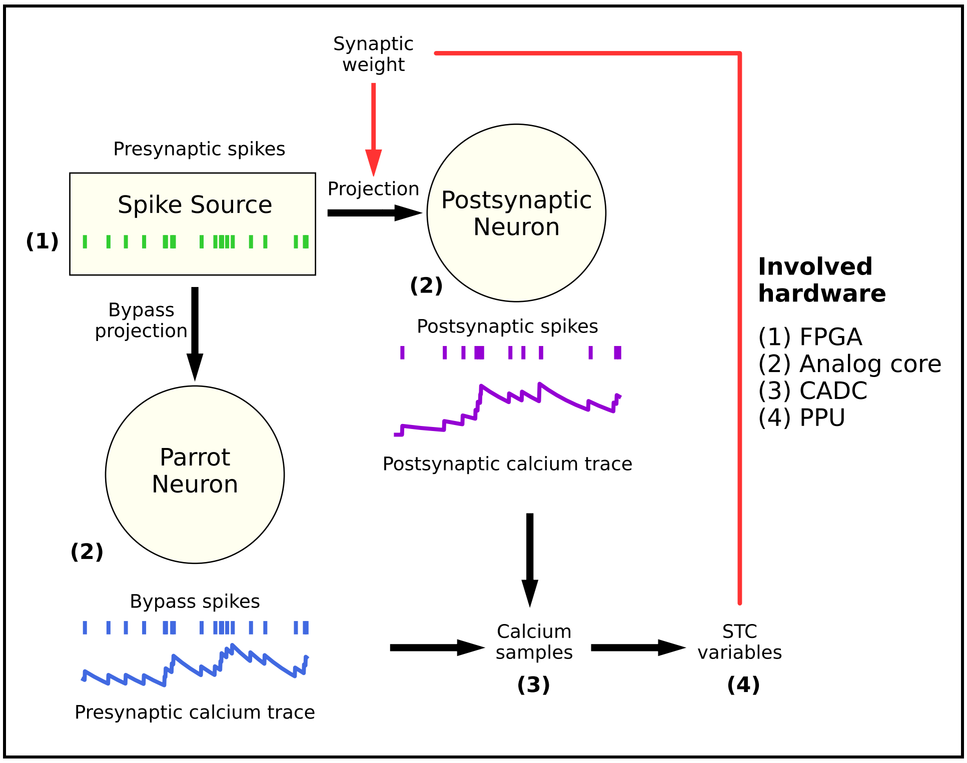


Figure 3.2: Experimental setup for emulating a single synapse following the STC plasticity rule on BrainScaleS-2. A spike source is used to generate presynaptic spikes. These are forwarded to the postsynaptic neuron via a projection whose weight update obeys the STC plasticity rule. These spikes are injected by the FPGA. The same presynaptic spikes are mirrored to the parrot neuron using a bypass projection to emulate the presynaptic calcium dynamics. The bypass mode is supported by the neurons of BrainScaleS-2 which ensures that presynaptic spikes are immediately translated to postsynaptic spikes. The neuron and calcium dynamics are emulated in the analog core of BrainScaleS-2. The weighted sum of the adaptation traces of the postsynaptic and parrot neurons represent the calcium trace. Calcium samples obtained by the CADC are then used to calculate the STC variables using the PPU including the synaptic weight which is used to update the weight of the projection.

3.1.4 Hardware Constraints and Limitations

For emulating the STC plasticity rule on BrainScaleS-2, there are several constraints to be accounted for:

1. PPU period: this period accounts for the time required by the PPU to sample from the analog traces and do the necessary calculations. In our application, we need to sample from the adaptation traces, calculate the state variables, update the synaptic weights, and write the final weight to the synaptic projection. The timestep used in the model simulation (Luboeinski and Tetzlaff, 2021) was 0.2 ms which translates to 0.2 µs when accounting for the acceleration factor. However, we already demonstrated in the lab rotation (Atoui, 2024) that this is not enough for the PPU, and we suggested using slower

time updates that must not exceed 100 µs and preferably remain below 50 µs to ensure the expected behavior of the stimulation protocols.

2. Using integers instead of floats: The PPU relies primarily on 8-bit and 16-bit integers, whereas the simulations were performed using double-precision arithmetic (Luboeinski and Tetzlaff, 2021). Though the PPU also supports floating-point calculations, we prioritize using 8-bit arithmetic for speed purposes as the STC plasticity rule imposes many calculations and the PPU period should remain low. This necessitates converting the STC variables to a suitable range, but also imposes truncation errors.

3. Floating-point operations: besides representing the STC variables as integers, the operations of these variables with STC parameters that are mostly decimal numbers are time-expensive, which increase the PPU period. This has to be resolved by further approximations of the STC parameters by writing them as fractions and performing integer operations only.

4. Large time constants: these lead to slow changes of STC variables such as the protein amount, the late-phase weight, and the steady-state term of the early-phase weight that cannot be represented by reduced-precision arithmetic. The problem arises due to the small range of 8-bit integers which can only represent numbers between 0 and 255. This leads to stagnation since the changes in the state variables and synaptic weights are small and cannot be represented by 8-bit integers. To resolve this, we suggest using a stochastic-update framework.

Aside from these constraints, there exist further limitations for emulating STC on BrainScaleS-2:

1. The postsynaptic calcium influx delay arising from the presynaptic spikes, described in eq. (3.4) as $t_{c,delay}$ will not be implemented. The reason is that the calcium traces are generated by the adaptation traces whose circuits are designed to increment the adaptation term at the spike time with no possibility for a predefined delay.

2. The axonal spike delay will not be implemented as the circuits for emulating neuron dynamics do not account for this delay.

3. The noise fluctuations of calcium during potentiation and depression that affect the early-phase weight, referred to as $\xi(t)$ described in eq. (3.5), will not be implemented as the implementation is computationally expensive.

3.2 Emulating Neuron Dynamics

To adjust the neuron dynamics to a desired behavior, each silicon neuron in BrainScaleS-2 has 24 uniquely tuneable analog parameters (Pehle et al., 2022). For the LIF neuron model, only 15 analog parameters out of the 24 parameters are relevant. Here, it is useful to note that the differential equation that governs the neural dynamics is invariant to some parameters. For example, the biological reset, reversal, and threshold membrane potentials are $-70\,\mathrm{mV}$, $-65\,\mathrm{mV}$ and $-55\,\mathrm{mV}$ respectively. In neuromorphic hardware, the priority is to optimize the design to the technology used in the manufacturing process, and this includes the design choices for the state variables such as the membrane and adaptation potentials. Eventually, the neural network behavior relies on spikes as the mode of communication between neurons, and the biological membrane potentials can be mapped to different hardware membrane potentials that can generate spikes in a similar fashion to the biological neuron. The relevant neuron parameters for our application that are directly accessible to the users of BrainScaleS-2 are listed below:

- **leak_v_leak**: leak membrane potential
- **leak_i_bias**: current that controls membrane resistance
- **threshold_v_threshold**: spiking threshold potential
- **reset_v_reset**: reset membrane potential after a spike
- **membrane_capacitance_capacitance**: membrane capacitance
- **refractory_period_refractory_time**: refractory period after a spike

Additionally, there are digital controls and analog parameters that control the synaptic current. The typical procedure followed here is to choose the highest value for the membrane capacitance to minimize noise on the membrane, as the membrane acts as a lowpass RC filter. Accordingly, the leak conductance can be calibrated to obtain the desired membrane time constant τ_m. The reset, leak, and threshold potentials are tuned to achieve a linear circuit operation with $V_{reset} < V_{leak} < V_{threshold}$. The most important consideration here is the integration from the leak potential to the threshold potential in the LIF model that occurs in the presence of postsynaptic potentials. This integration should be mapped to that in the biological model to produce the same number of spikes at the same time instants. A calibration of the threshold potential and the synaptic input current by analog parameters is thus required to obtain the desired spikes.

Practically, the weight update will not affect the performance of the single-synapse emulation since postsynaptic spikes occur only for the strong tetanic stimulation at a very low rate (Atoui, 2024). Therefore, the calibration of the threshold potential and synaptic input will not affect the emulation of the STC plasticity rule on a single synapse on BrainScaleS-2. For time constraints in this thesis, we chose to postpone this step for later stages where the weight update on hardware is required for the network emulation and affects the network's performance.

3.3 Emulating Calcium Dynamics

The neuron circuits of BrainScaleS-2 were designed and implemented in a way that ensures a wide flexibility in their usage, and this is made accessible to the user through many parameters that can be tailored to the application at hand. To emulate the calcium dynamics for the STC rule, we need a calcium trace that depends on the presynaptic and postsynaptic spikes as in equation eq. (3.4) without affecting the membrane potential and the neuron dynamics. More specifically, a circuit that implements the exponential kernel at each spike is required, and this can be achieved by the circuit of the adaptation current whose dynamics provided in eq. (2.5) can be easily compared to the calcium dynamics. In the following, we will describe how the calcium trace was obtained using the adaptation circuit of BrainScaleS-2 while achieving the same biological outcome as the STC model.

3.3.1 Tuning Hardware Parameters

To understand how the calcium trace can be implemented by the adaptation circuit, we can refer to the AdEx model and its parameters. The subthreshold adaptation strength a couples the adaptation current I_{adapt} with the membrane potential. This is not required in the calcium dynamics in equation 3.4 since the calcium concentration does not depend on the membrane potential. The time constant τ_{adapt} is mapped to the calcium time constant τ_c. The spike-triggered increment b controls the increment of the adaptation current at each spike which should be mapped to the contributions of presynaptic and postsynaptic spikes c_{pre} and c_{post} respectively in the calcium trace. An important consideration is that b should also be tuned to ensure a high dynamic range with a linear operation of the circuit. Here, we mention the main parameters that allowed decoupling the adaptation current from the membrane potential and tuning its parameters to resemble the biological calcium trace in both, the parrot and postsynaptic neurons:

- **adaptation_enable_pulse**: this parameter enables the usage of the adaptation circuit, set to *True* as we will rely on the adaptation term of the AdEx model to emulate the calcium dynamics.
- **adaptation_enable**: this parameter applies the adaptation current to the dynamics of the membrane potential of the neuron, set to *False* so as to decouple the adaptation trace from the membrane potential as the calcium concentration should not directly affect the neuron dynamics.
- **adaptation_i_bias_tau**: this is a conductance parameter that controls the time constant of the adaptation term. Note that the calcium time constant should be divided by the acceleration factor, i.e. 1000, so this parameter is mapped to $\tau_c = 48.8\,\mu\text{s}$. It is fine-tuned according to the chip and interaction with other parameters.
- **adaptation_i_bias_a**: this is a conductance parameter that controls the strength

of the subthreshold adaptation term a in eq. (2.5), set to 0 since the calcium concentration does not depend on the membrane potential.

- **adaptation_i_bias_b**: this is a conductance parameter that controls the spike-triggered adaptation term b in eq. (2.5). In our application, it controls the increment in the calcium trace at each spike. This parameter is fine-tuned to achieve an adequate dynamic range of calcium. It should not be too small so that noise does not affect small calcium readings, but also not too large so that the calcium region of interest remains within the linear operating region to map thresholds linearly as will be demonstrated later (Billaudelle, 2022). It is fine-tuned according to the chip and interaction with other parameters.
- **adaptation_v_ref**: this is the baseline voltage of the adaptation term, it is selected (Billaudelle, 2022) and fine-tuned to have the adaptation term operate in a linear range to obtain a smooth calcium trace. It is fine-tuned according to the chip and interaction with other parameters.

3.3.2 Obtaining Calcium Traces

For tuning the calcium parameters listed in section 3.3.1 and comparing against the true biological calcium trace, we rely on the MADC. Although the MADC will not be used for sampling the calcium traces when running the STC rule on BrainScaleS-2, it has a higher sampling rate and a finer resolution compared to the CADC. Therefore, the full adaptation traces can be retrieved, and the parameters can be fine-tuned to match the true calcium trace. We use the experimental setup described in fig. 3.2 and a stimulation scheme of Poisson spikes at 2 frequencies, 20 kHz and 100 kHz since these values match the frequencies used in the stimulation protocols that would be described in the following section.

3.3.3 Sampling Calcium Traces

While running the STC plasticity rule, the calcium traces will be sampled using the CADC at lower sampling rate and resolution compared to the MADC. Therefore, it is essential to look at the retrieved samples and compare them to the true biological traces to check beforehand if the experimental samples can track the true traces and fine-tune the parameters if necessary. Similar to section 3.3.2, we use the experimental setup described in fig. 3.2 and a stimulation scheme of Poisson spikes at the 2 frequencies, 20 kHz and 100 kHz. Here we also note from previous experiments on BrainScaleS-2 that there exists a variability in the sampling times in the order of microseconds that might affect the samples relative to the expected traces.

3.3.4 Mapping Calcium Thresholds

Referring to the early-phase weight equation eq. (3.5), the calcium concentration is used in a form of thresholds for updating the early-phase weight. This provides the advantage of having to map only the potentiation and depression thresholds without the need to map the full calcium range. In fact, mapping the full calcium range is challenging as the adaptation circuit becomes nonlinear for high calcium values, and noise can interfere with low calcium values. Here we describe the approach used to map the two thresholds θ_p and θ_d.

Calcium Mapping Protocol

It is useful to define a stimulation protocol for calcium mapping that we can understand its behavior beforehand and design according to our specifications. For that, we use spikes with a fixed interspike interval (ISI). Here we use lowercase letter c for biological calcium and uppercase C for calcium on hardware with LSB units. For simplicity, we now only use the calcium concentration arising from presynaptic spikes which reduces equation 3.4 to:

$$\frac{dc(t)}{dt} = -\frac{c(t)}{\tau_{\mathrm{c}}} + c_{\mathrm{pre}} \sum_n \delta(t - t^n) \tag{3.9}$$

The analytical solution for eq. (3.9) is:

$$c(t) = c_{\mathrm{pre}} \sum_{n=0}^{N-1} \exp\,(-\frac{t - t^n}{\tau_c}) \cdot \Theta(t - t^n) \tag{3.10}$$

Now we consider spikes separated by a fixed ISI, τ_{isi}, so eq. (3.10) can be further simplified to:

$$c(t) = \sum_{n=0}^{N-1} \exp\,(-\frac{t - n \cdot \tau_{isi}}{\tau_c}) \cdot \Theta(t - n \cdot \tau_{isi}) \tag{3.11}$$

For $t \geq t^N$, and using the sum of a geometric sequence, the calcium concentration can be written as:

$$\begin{aligned} c(t) &= \sum_{n=0}^{N-1} \exp\,(-\frac{t - n \cdot \tau_{isi}}{\tau_c}) = \exp\,(-\frac{t}{\tau_c}) \cdot \sum_{n=0}^{N-1} (\exp\,(\frac{\tau_{isi}}{\tau_c}))^n \\ &= \exp\,(-\frac{t}{\tau_c}) \cdot \frac{\exp\,(\frac{N \cdot \tau_{isi}}{\tau_c}) - 1}{\exp\,(\frac{\tau_{isi}}{\tau_c}) - 1} \end{aligned} \tag{3.12}$$

At the time of the $(N-1)^{th}$ spike,

$$c(t = (N-1) \cdot \tau_{isi}) = \frac{\exp\,(\frac{\tau_{isi}}{\tau_c}) - \exp\,(\frac{-(N-1) \cdot \tau_{isi}}{\tau_c})}{\exp\,(\frac{\tau_{isi}}{\tau_c}) - 1} \tag{3.13}$$

which has an upper bound for a large N

$$c_{max} = c(t = (N-1) \cdot \tau_{isi}) \longrightarrow \frac{\exp\left(\frac{\tau_{isi}}{\tau_c}\right)}{\exp\left(\frac{\tau_{isi}}{\tau_c}\right) - 1} \tag{3.14}$$

For our mapping protocol, the maximum calcium concentration c_{max} reached should be within the linearity region. The biological calcium concentration can be attributed to the number of spike-triggered increments performed by the adaptation circuit. Using the design and measurement curves in (Billaudelle, 2022) as an average among different chips, we can set $c_{max} = 5$, i.e. the adaptation term can be incremented by 5 without leaving the linear operating region.

Using eq. (3.14), τ_{isi} can be computed as:

$$\tau_{isi} = \tau_c \cdot \ln\left(1 + \frac{1}{c_{max} - 1}\right), \tag{3.15}$$

which amounts to 10.9 µs with $c_{max} = 5$ and $\tau_c = 0.0488$ ms.

To collect samples that cover the calcium range up to $c_{max} = 5$, we use a stimulation protocol for a predefined total duration. The duration is divided into periods where the first period includes only 1 spike, and the number of spikes increases at a rate of 1 spike/period (see figure 3.3). We use such a stimulation protocol to have a wide range of biological calcium values and account for unexpected hardware behavior. Within each period, there will be an interval of spikes and an interval with no spikes where the calcium concentration decays. To avoid the expected error from the variability in the sampling times around the rising edges of spikes, we only consider the samples that are in the decaying interval and leave out the samples taken within the interval of spikes.

Linear Regression

The approximately linear operation of the circuit responsible for emulating the calcium dynamics up to $c_{max} = 5$ allows us to formulate a simple linear regression problem to map the calcium potentiation and depression thresholds, $\theta_p = 3$ and $\theta_d = 1.2$ respectively, to the hardware readings. Specifically, let C_h be the hardware readings of the analog circuit, C_b be the biological calcium concentration, β_0 be the regression coefficient, and β_1 be the regression constant which is the calcium baseline:

$$C_h = \beta_0 \cdot C_b + \beta_1. \tag{3.16}$$

β_0 and β_1 can be estimated using the method of least-squares estimation that minimizes the squared discrepancies between the observed data C_h and the expected data C_b (Hastie et al., 2001). Using the estimated parameters, we get:

$$\theta_{p(emulation)} = \hat{\beta}_0 \cdot \theta_p \tag{3.17}$$

and

$$\theta_{d(emulation)} = \hat{\beta}_0 \cdot \theta_d \tag{3.18}$$

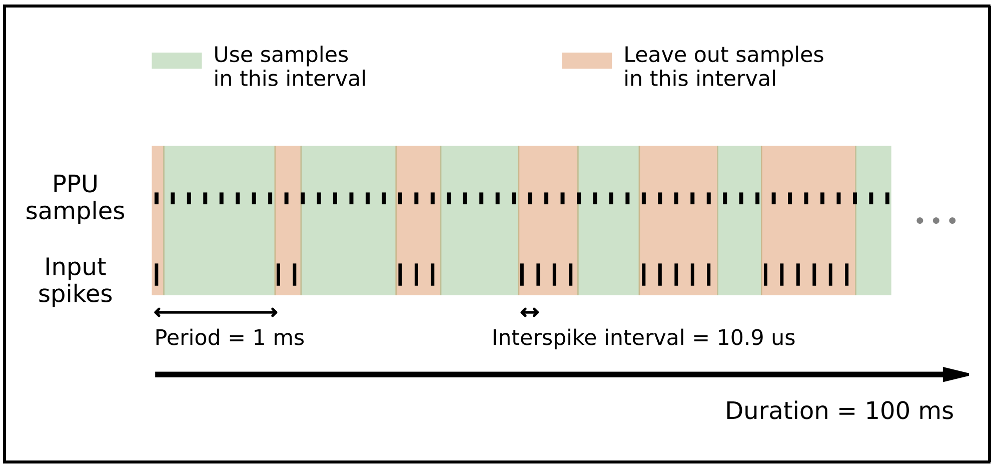

Use samples in this interval
Leave out samples in this interval
PPU samples
Input spikes
Period = 1 ms
Interspike interval = 10.9 us
Duration = 100 ms

Figure 3.3: Stimulation protocol used for mapping calcium thresholds. The full duration of the protocol is divided into multiple periods. Each period consists of injecting spikes separated by a fixed ISI starting with 1 spike and increasing at a rate of 1 spike/period. PPU samples that are within the interval of spike injection are left-out due to possible errors from the variability in the sampling times. PPU samples that are within the decaying interval of calcium are selected for threshold mapping. The timing in the figure is not up to scale.

with the calcium baseline being measured before every run of the plasticity rule and subtracted from the measured calcium. It is worthy to note the variability of these parameters between different neurons on the same chip and between different chips, so this mapping should be repeated whenever we use different neurons or chips.

3.4 Emulating the Synaptic Plasticity Rule

3.4.1 Plasticity Experiments

To model the different forms of plasticity that could occur in a single synapse, we use the standard stimulation protocols that are generally adopted in the literature. These protocols vary by strength and frequency as described in fig. 3.4. Tetanic (high-frequency) stimulations induce LTP due to the high calcium concentration released at the synapse while low-frequency stimulations induce LTD due to the moderate calcium concentration released at the synapse. Furthemore, strong stimulations cause the synapse to be marked by a local synaptic tag and the synthesis of a sufficient amount of PRPs necessary for late-phase LTM whereas weak stimulations only induce synaptic tagging necessary for early LTM . Consequently, the strong tetanic stimulation (STET) protocol induces early-phase and late-phase LTP whereas weak tetanic stimulation (WTET) protocol induces only early-phase LTP. On the other hand, strong low-frequency stimulation (SLFS) protocol induces early-phase and late-phase LTD whereas weak low-frequency stimulation (WLFS) protocol induces only early-phase LTD.

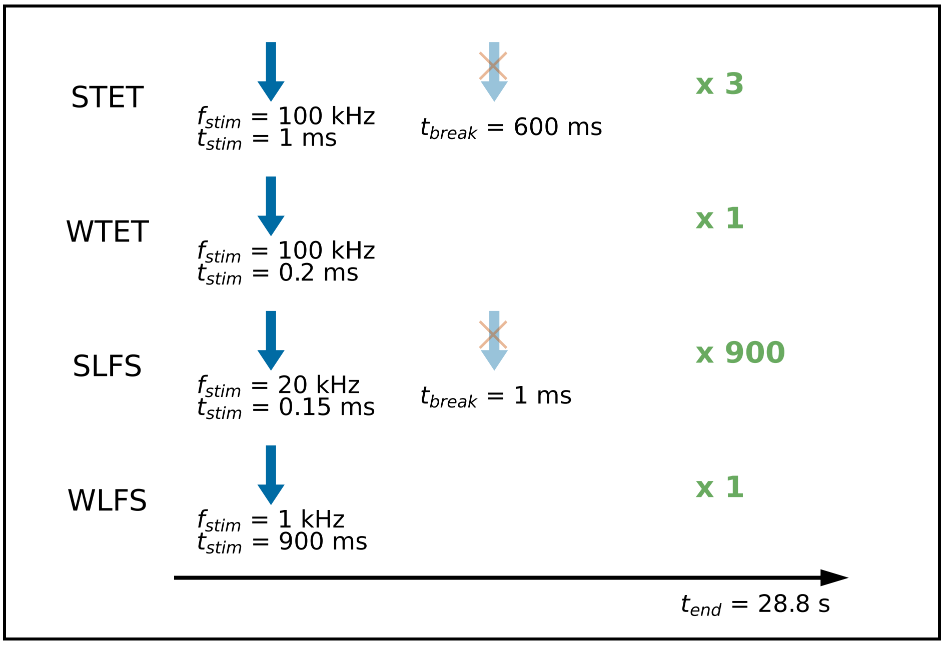


Figure 3.4: Standard plasticity protocols for the induction of early-phase and late-phase synaptic potentiation and depression. Reproduced from supplementary information of Luboeinski and Tetzlaff (2021) to match the timescale of BrainScaleS-2.

3.4.2 Mapping Biological Parameters to BrainScaleS-2

This step is needed since only integers will be used in the STC emulation on BrainScaleS-2 contrary to the original model defined in (Luboeinski and Tetzlaff, 2021) where decimal numbers and double precision arithmetic are adopted for the parameters and variables respectively. All the neuron and synapse parameter values that are used in the emulation scheme are summarized in table 3.2. In this section, we explain in detail our mapping choices.

Early-phase weight

Referring to eq. (3.5), the value of the early-phase weight h ranges between a minimum of $0\,\mathrm{nC}$ and a maximum of $1\,\mathrm{nC}$. Since we'll be using an 8-bit unsigned integer for the emulation of h, the maximum value of 1 can be mapped to $2^8 - 1 = 255$. The resolution of h is thus $\frac{1\,\mathrm{nC}}{255}$, and h_0 can be mapped to $0.420\,075\,\mathrm{nC} \cdot 255\,\mathrm{LSB/nC} = 107\,\mathrm{LSB}$. The potentiation, depression, and steady-state factors, γ_p, γ_d, and 0.1 respectively, are held constant as they are unitless multiplicative factors of the time constant τ_h. On the other hand, τ_h is divided by the acceleration factor of 1000. The potentiation and depression calcium thresholds, θ_p and θ_d respectively, are mapped to the range of calcium as per the methods described in section 3.3.4, and the resulting values expressed in units of LSB and presented in section 4.1.3 are used.

Protein amount

The protein amount p will be represented as an 8-bit unsigned integer with range 0 till 255. Referring to eq. (3.6), the time constant τ_p is divided by the acceleration factor of 1000, and the protein synthesis threshold θ_{pro} is mapped to the range of h which yields $\theta_{pro} = 0.210\,037 \cdot 255 = 54\,\mathrm{LSB}$. To map the protein synthesis rate α, we can look at the solution of the differential equation. When the protein synthesis condition is satisfied, eq. (3.6) is expressed as:

$$\tau_p \frac{dp(t)}{dt} = -p(t) + \alpha, \tag{3.19}$$

whose solution with an initial amount of protein p_0 can be expressed as:

$$p(t) = \alpha + (p_0 - \alpha)\exp(-\frac{t}{\tau_p}), \tag{3.20}$$

which has a limit of α as $t \longrightarrow \infty$. To maximize the dynamic range of the protein amount on BrainScaleS-2, we choose $\alpha^{hw} = 255$.

Late-phase weight

Referring to eq. (3.7), the late-phase weight z ranges between -0.5 and 1, so we use a signed 8-bit integer to represent z. To ensure balanced stepsize for potentiation and depression, we map $z = 1$ to 127, $z = 0.5$ to 64, and $z = -0.5$ to -64 so the resolution is $\frac{1}{127}$. The time constant τ_z is divided by the acceleration factor 1000. The tagging threshold θ_{tag} is mapped to the range of h, so $\theta_{tag} = 0.0840149 \cdot 255 = 21$ LSB. Finally, the protein amount in the equation is normalized by 255.

Total synaptic weight

The model by Luboeinski and Tetzlaff (2021) models plasticity for excitatory synapses only, so the total synaptic weight is represented by an 8-bit unsigned integer with a range of 0 to 255. Referring to eq. (3.8), the term h_0 is used in its normalized form. To be able to add the early-phase weight h and the late-phase weight z, they should have the same resolution, so the late-phase weight is multiplied by a factor of 2.

Symbol	Simulation Value	Emulation Value	Remarks on Mapping
Δt_V	$0.2\,\text{ms}$	0	Continuous-time emulation of neuron dynamics
Δt_C	$0.2\,\text{ms}$	0	Continuous-time emulation of calcium dynamics
$\Delta t_{weights}$	$0.2\,\text{ms}$	$50\,\mu\text{s}$	Acceleration factor, additional time for PPU
t_{end}	8 hours	$28.8\,\text{s}$	Acceleration factor
τ_{mem}	$10\,\text{ms}$	$10\,\mu\text{s}$	Acceleration factor
τ_{syn}	$5\,\text{ms}$	$5\,\mu\text{s}$	Acceleration factor
$t_{ax,delay}$	$3\,\text{ms}$	N/A	Limitation of neuron circuit
t_{ref}	$2\,\text{ms}$	$2\,\mu\text{s}$	Acceleration factor
R	$10\,\text{M}\Omega$	$4\,\text{M}\Omega$	Circuit operation convenience
V_{rev}	$-65\,\text{mV}$	70 LSB	Circuit operation convenience
V_{reset}	$-70\,\text{mV}$	65 LSB	Circuit operation convenience
V_{th}	$-55\,\text{mV}$	85 LSB	Circuit operation convenience
h_0	$0.420\,075\,\text{nC}$	107 LSB	Mapping to integers
$t_{c,delay}$	$0.0188\,\text{s}$	N/A	Limitation of neuron circuit
c_{pre}	1	1	Multiplicative factor
c_{post}	0.2758	0.2758	Multiplicative factor
τ_c	$0.0488\,\text{s}$	$0.0488\,\text{ms}$	Acceleration factor
τ_p	$3600\,\text{s}$	$3600\,\text{ms}$	Acceleration factor
τ_z	$3600\,\text{s}$	$3600\,\text{ms}$	Acceleration factor
γ_p	1645.6	1645.6	Multiplicative factor
γ_d	313.1	313.1	Multiplicative factor
θ_p	3	52*	Experimental mapping
θ_d	1.2	20*	Experimental mapping
σ_{pl}	$0.290\,436\,\text{nC}\,\text{s}^{-1/2}$	N/A	Computational convenience
α	1	255	Mapping to integers
θ_{pro}	$0.210\,037\,\text{nC}$	54 LSB	Mapping to integers
θ_{tag}	$0.084\,014\,9\,\text{nC}$	21 LSB	Mapping to integers

Table 3.2: Neuron and synapse model parameters for simulation versus emulation schemes. The simulation values are adapted from Luboeinski and Tetzlaff (2021). The neuron membrane potential and calcium concentration are continuous-time signals emulated on the analog core. The parameters concerned with time dynamics are divided by the acceleration factor 1000. The multiplicative factors are kept constant. The thresholds and initial or steady-state values are mapped to the ranges of their corresponding variables. The LSB units are units of measurement of the CADC. Parameters that are not accounted for in the emulation are referred to as not applicable (N/A). Parameters with (*) are chip-dependent.

3.4.3 Implementing the Plasticity Rule

Solving the Differential Equations

To solve the differential equations of h, p, and z in the plasticity kernel, we use the explicit Euler method to update the STC variables (Kong et al., 2021). More specifically, let $\frac{dy(t)}{dt} = F(t, y)$ be a first-order differential equation. The linear approximation of $y(t)$ at t_{n+1} around t_n is:

$$y(t_{n+1}) = y(t_n) + (t_{n+1} - t_n) \cdot \left.\frac{dy}{dt}\right|_{t=t_n} \tag{3.21}$$

For a regular time step $\Delta t = t_{n+1} - t_n$, we can re-write the explicit Euler formula as:

$$y(t_n + \Delta t) = y(t_n) + \Delta t \cdot F(t_n, y(t_n)) \tag{3.22}$$

Stochastic Rounding

As demonstrated in section 2.4, the rounding error is higher when using the RN rounding scheme compared to the SR rounding scheme. To decrease the rounding errors when multiplying integers with fractions and the accumulated errors in operations dependent on the rounded result, we rely on the SR rounding scheme for integer multiplication with fractions. Specifically, let x be the result of the multiplication of an integer with a fraction such that $x \in \mathbb{Q}$, so that $\lfloor x \rfloor$ and $\lceil x \rceil$ are the two nearest numbers to x in $\mathbb{Z}$, and $\epsilon_M = 1$. The SR rounding scheme for x becomes:

$$\mathrm{SR}(x) = \begin{cases} \lfloor x \rfloor \text{ if } P > x - \lfloor x \rfloor \\ \lceil x \rceil \text{ if } P \leq x - \lfloor x \rfloor \end{cases} \tag{3.23}$$

where P is a random number drawn from a uniform distribution $\mathbb{U}(0, 1)$.

Stochastic Updates

To overcome the phenomenon of stagnation demonstrated in section 2.4 when updating the STC variables, we use SR in solving the corresponding differential equations. More specifically, let x be a variable that is updated by eq. (3.22) using the following update rule:

$$x(t + \Delta t) = x(t) + k \cdot \Delta t, \tag{3.24}$$

where k is a constant such that $k \cdot \Delta t < 1$. In this case, we define the SR rounding scheme of the update rule as follows:

$$x(t + \Delta t) = \begin{cases} x(t) + 1 \text{ if } P \leq k \cdot \Delta t \\ x(t) \text{ if } P > k \cdot \Delta t, \end{cases} \tag{3.25}$$

where P is a random number drawn from a uniform distribution $\mathbb{U}(0, 1)$. In other words, x is updated effectively with a probability of $k \cdot \Delta t$ such that on average, the variables are updated at a timestep of Δt. Although this update scheme is a form of SR, we refer to it as stochastic update (SU) to differentiate the two goals of rounding in our application. It is worth to note that we do not intend to increase the update timestep Δt and perform regular updates as this yields highly distorted results as demonstrated in the lab rotation (Atoui, 2024).

Algorithm Implementation

Using eq. (3.22) and section 3.4.2, the differential equation for the early-phase weight h in eq. (3.5) can be rewritten as:

$$\begin{aligned} h(t+\Delta t) = h(t) &+ \frac{0.1 \cdot \Delta t}{\tau_h}\,(h_0 - h(t)) \\ &+ \frac{\gamma_{\mathrm{p}} \cdot \Delta t}{\tau_h} \cdot (255 - h(t)) \cdot \Theta[c(t) - \theta_{\mathrm{p}}] \\ &- \frac{\gamma_{\mathrm{d}} \cdot \Delta t}{\tau_h} \cdot h(t) \cdot \Theta[c(t) - \theta_{\mathrm{d}}], \end{aligned} \tag{3.26}$$

The variety in the early-phase weight dynamics allows us to divide the equation into three parts. The first two parts are related to plasticty in the presence of sufficient calcium while the third part always pull the early-phase weight to a steady-state value.

1. Plasticity cases

   Case of LTP: $c \geq \theta_p > \theta_d$ yields:

   $$h(t+\Delta t) = h(t) \cdot (1 - \frac{\Delta t}{\tau_h} \cdot (\gamma_{\mathrm{p}} + \gamma_{\mathrm{d}})) + \frac{\gamma_{\mathrm{p}} \cdot \Delta t \cdot 255}{\tau_h}, \tag{3.27}$$

   Using the values in table 3.2, the early-phase weight can be updated at each timestep by applying SR as in section 3.4.3 when multiplying with the fraction $(1 - \frac{\Delta t}{\tau_h} \cdot (\gamma_{\mathrm{p}} + \gamma_{\mathrm{d}}))$ and adding a constant offset $\frac{\gamma_{\mathrm{p}} \cdot \Delta t \cdot 255}{\tau_h}$.

   $$h(t+\Delta t) = \begin{cases} \lceil h(t+\Delta t) \rceil \text{ if } P_h \leq h(t+\Delta t) - \lfloor h(t+\Delta t) \rfloor \\ \lfloor h(t+\Delta t) \rfloor \text{ if } P_h > h(t+\Delta t) - \lfloor h(t+\Delta t) \rfloor \end{cases} \tag{3.28}$$

   where P_h is a random number drawn from a uniform distribution $\mathbb{U}(0, 1)$.

   Case of LTD: $\theta_p > c \geq \theta_d$ yields:

   $$h(t+\Delta t) = h(t) \cdot (1 - \frac{\gamma_{\mathrm{d}} \cdot \Delta t}{\tau_h}), \tag{3.29}$$

   Here, the early-phase weight can be updated at each timestep by applying SR described in section 3.4.3 when multiplying with the fraction $(1 - \frac{\gamma_{\mathrm{d}} \cdot \Delta t}{\tau_h})$ similar to eq. (3.28).

2. Steady-state: this part of the equation forces h to converge to a steady-state value h_0. The steady-state dynamics are slow with a time constant of $10 \cdot \tau_h$. In this case, we apply SU proposed in section 3.4.3 as follows:

$$h(t+\Delta t) = \begin{cases} h(t) + 1 \cdot \operatorname{sgn}(h_0 - h(t)) \text{ if } P_{h,ss} \leq \dfrac{0.1\Delta t \cdot |h_0 - h(t)|}{\tau_h} \\ h(t) \text{ if } P_{h,ss} > \dfrac{0.1\Delta t \cdot |h_0 - h(t)|}{\tau_h} \end{cases} \tag{3.30}$$

with $P_{h,ss}$ being a random number drawn from a uniform distribution $\mathbb{U}(0,1)$. We chose $P_{h,ss}$ to be dependent on $h_0 - h(t)$ so that $h(t)$ converges smoothly.

Using eq. (3.22) and section 3.4.2, the differential equation for the protein amount p in eq. (3.6) can be rewritten as:

$$p(t+\Delta t) = p(t) - p(t) \cdot \frac{\Delta t}{\tau_p} + \frac{\alpha \cdot \Delta t}{\tau_p} \cdot\ \Theta[(|h(t) - h_0|) - \theta_{\text{pro}}] \tag{3.31}$$

Since $\frac{\Delta t}{\tau_p} < 1$ and $\frac{\alpha \cdot \Delta t}{\tau_p} < 1$, the evolution of the protein dynamics can be divided into 2 parts that exploit SU:

1. Case of protein synthesis during LTM: $|h(t) - h_0| > \theta_{pro}$ yields:

$$p(t+\Delta t) = \begin{cases} p(t) + 1 \text{ if } P_p \leq \dfrac{\alpha \cdot \Delta t}{\tau_p} \\ p(t) \text{ if } P_p > \dfrac{\alpha \cdot \Delta t}{\tau_p} \end{cases} \tag{3.32}$$

with P_p being a random number drawn from a uniform distribution $\mathbb{U}(0,1)$.

2. Steady-state: this part of the equation allows the protein amount to decay back to 0 which yields:

$$p(t+\Delta t) = \begin{cases} p(t) - 1 \text{ if } P_{p,ss} \leq p(t) \cdot \dfrac{\Delta t}{\tau_p} \\ p(t) \text{ if } P_{p,ss} > p(t) \cdot \dfrac{\Delta t}{\tau_p} \end{cases} \tag{3.33}$$

where $P_{p,ss}$ is a random number drawn from a uniform distribution $\mathbb{U}(0,1)$. We chose $P_{p,ss}$ to be dependent on $p(t)$ so that $p(t)$ decays smoothly and more frequently.

Using eq. (3.22) and section 3.4.2, the differential equation for the late-phase weight z in eq. (3.7) can be rewritten as:

$$\begin{aligned} z(t+\Delta t) = z(t) + \frac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (127 - z(t)) \cdot \Theta[(h(t) - h_0) - \theta_{\text{tag}}] \\ - \frac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (z(t) + 64) \cdot \Theta[(h_0 - h(t)) - \theta_{\text{tag}}] \end{aligned} \tag{3.34}$$

The evolution of the late-phase weight dynamics follows 2 cases:

1. Case of late LTP: $h(t) - h_0 \geq \theta_{\text{tag}}$ yields:

$$z(t+\Delta t) = \begin{cases} z(t) + 1 \text{ if } P_z \leq \dfrac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (127 - z(t)) \\ z(t) \text{ if } P_z > \dfrac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (127 - z(t)) \end{cases} \tag{3.35}$$

   Furthermore, since z depends directly on p, it makes sense that the update probability of z is directly dependent on the update probability of p. This yields:

$$z(t+\Delta t) = \begin{cases} z(t) + 1 \text{ if } P_z \leq \dfrac{p(t)}{255 \cdot \alpha} \cdot (127 - z(t)) \text{ if } P_p \leq \dfrac{\alpha \cdot \Delta t}{\tau_p} \\ z(t) \text{ if } P_z > \dfrac{p(t)}{255 \cdot \alpha} \cdot (127 - z(t)) \text{ if } P_p \leq \dfrac{\alpha \cdot \Delta t}{\tau_p} \end{cases} \tag{3.36}$$

2. Case of late LTD: $h_0 - h(t) \geq \theta_{\text{tag}}$ yields:

$$z(t+\Delta t) = \begin{cases} z(t) - 1 \text{ if } P_z \leq \dfrac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (z(t) + 64) \\ z(t) \text{ if } P_z > \dfrac{p(t) \cdot \Delta t}{255 \cdot \tau_z} \cdot (z(t) + 64) \end{cases} \tag{3.37}$$

   Similar to the case of LTP, we let the update probability of z be directly dependent on that of p which yields:

$$z(t+\Delta t) = \begin{cases} z(t) - 1 \text{ if } P_z \leq \dfrac{p(t)}{255 \cdot \alpha} \cdot (z(t) + 64) \text{ if } P_p \leq \dfrac{\alpha \cdot \Delta t}{\tau_p} \\ z(t) \text{ if } P_z > \dfrac{p(t) \cdot \Delta t}{255 \cdot \alpha} \cdot (z(t) + 64) \text{ if } P_p \leq \dfrac{\alpha \cdot \Delta t}{\tau_p} \end{cases} \tag{3.38}$$

where P_z is a random number drawn from a uniform $\mathbb{U}(0, 1)$. We chose P_z to be dependent on $z(t)$ so that $z(t)$ converges smoothly and more frequently. Finally, the total synaptic weight w can be updated as follows:

$$w(t+\Delta t) = h(t+\Delta t) + 2 \cdot h_0 \cdot z(t+\Delta t) \tag{3.39}$$

where h_0 is the normalized steady-state early-phase weight value.

The arithmetic representation of the STC variables in the plasticity kernel and their update routine are summarized in table 3.3.

To implement the SU routine, we rely on random 32-bit unsigned integers drawn from a 32-bit xorshift pseudo-random number generator (Marsaglia, 2003). The update probabilities are then converted to integers by multiplying each probability with $2^{32} - 1$ at runtime to obtain *probability-equivalents*, and the 32-bit random numbers are compared against these probability-equivalents. Furthermore, we represent any decimal number x using a fraction $\frac{n}{d}$ such that $|x - \frac{n}{d}| < \epsilon$ with ϵ being a predefined error parameter set to 0.001.

STC Variable	Datatype	Resolution	Cases	Update Probability
Early-phase weight h	8-bit unsigned integer	$\frac{1}{255}$	Early LTP	1
			Early LTD	1
			Steady-state	$\frac{\lvert h(t) - h_0 \rvert}{125164}$
Protein amount (p)	8-bit unsigned integer	$\frac{1}{255}$	Protein synthesis	$\frac{1}{257}$
			Protein decay	$\frac{p(t)}{65455}$
Late-phase weight (z)	8-bit signed integer	$\frac{2}{255}$	Late LTP	$\frac{p(t) \cdot (127 - z(t))}{16690909}$
			Late LTD	$\frac{p(t) \cdot (z(t) + 64)}{16690909}$
Total synaptic weight (w)	8-bit unsigned integer	$\frac{1}{255}$	LTM	1

Table 3.3: Summary of the STC variables updated on the PPU with their properties and update probability. For each update case, the update probability is evaluated at $\Delta t = 50$µs. Variables that are updated at every timestep have an update probability of 1.

Algorithm 1: STC Plasticity Kernel

```
Result: $h, p, z, w$
$C_{baseline,pre} \leftarrow 255$;
$C_{baseline,post} \leftarrow 255$;
while $t \leq t_{end}$ do
    while $t \leq t_{baseline}$ do
        $C_{pre}, C_{post} \leftarrow \text{AnalogReadOut}()$;
        $C_{baseline,pre} \leftarrow \min(C_{baseline,pre}, C_{pre}) - \text{ErrorMargin}$;
        $C_{baseline,post} \leftarrow \min(C_{baseline,post}, C_{post}) - \text{ErrorMargin}$;
    end
    $C_{pre}, C_{post} \leftarrow \text{AnalogReadOut}()$;
    $C \leftarrow k_{pre} \cdot (C_{pre} - C_{baseline,pre}) + k_{post} \cdot (C_{post} - C_{baseline,post})$;
    if $C \geq \theta_p$ then
        $h \leftarrow h \times \text{factor}_\text{p} + \text{offset}_\text{p}$;
    else
        if $C \geq \theta_d$ then
            $h \leftarrow h \times \text{factor}_\text{d}$;
        end
    end
    if $P_h \leq h - \lfloor h \rfloor$ then
        $h \leftarrow \lceil h \rceil$
    end
    if $P_{h,ss} \leq \frac{0.1\Delta t \cdot |h_0 - h|}{\tau_h}$ then
        $h \leftarrow h + 1 * \text{sgn}(h_0 - h)$;
    end
    if $P_p \leq \frac{\alpha \cdot \Delta t}{\tau_p}$ then
        if $|h - h_0| \geq \theta_{pro}$ then
            $p \leftarrow p + 1$;
        end
        if $(h - h_0) \geq \theta_{tag}$ and $P_z \leq \frac{p}{255 \cdot \alpha} \cdot (127 - z)$ then
            $z \leftarrow z + 1$;
        else
            if $(h_0 - h) \geq \theta_{tag}$ and $P_z \leq \frac{p}{255 \cdot \alpha} \cdot (z + 64)$ then
                $z \leftarrow z - 1$;
            end
        end
    end
    $w \leftarrow h + 2 * h_0 * z$;
    RecordResults();
end
```

3.5 Running Experiments on BrainScaleS-2

To assess our implementation of the STC plasticity rule, we run algorithm 1 at an update timestep of 0.05 ms for the four plasticity protocols on BrainScaleS-2. We compare the emulation results against two baselines obtained by simulation: the first is obtained at the base timestep of 0.2 ms biological time as in Luboeinski and Tetzlaff (2021), and the second is obtained at 50 ms which is the biological equivalent of the update timestep used in the emulation. The simulations were already reproduced in the lab rotation (Atoui, 2024), and a thorough assessment was done on the effect of increasing the simulation timestep for regular updates, but we reproduce them here only for comparison purposes with the emulation results. Luboeinski and Tetzlaff (2021) accounted only for the variability in spikes, as their implementation was based on a simulation on a standard computer. For the emulation on BrainScaleS-2, there are different sources of variability:

- Trial-to-trial variability: we repeat the experiment using the same set of spikes and update seed for 100 times to assess the variability and reproducibility of the same result. As far as our implementation in this work is concerned, the trial-to-trial variability stems from the emulation of calcium dynamics in the analog core of BrainScaleS-2 and sampling variability.
- Variability across update seeds: we use 100 different update seeds for the same set of spikes to assess our implementation and how close the average behavior across different update seeds is to the baseline obtained by simulation.
- Variability across different set of spikes: we vary the set of spikes using different seeds. Here we assess whether we can rely 100 set of spikes as in the simulation of Luboeinski and Tetzlaff (2021), or if we need more experiment runs to obtain significant results.
- Variability across different chips: we use 3 different chips to assess the similarity of the results across different neurons and whether the mismatches in the manufacturing process affect the results.
- Variability in the time samples: the time samples are not exactly taken at multiples of the predefined PPU period, which causes different results between the emulation at 50 µs and the simulation at the equivalent biological timestep 50 ms.

As quantitative measures of our implementation, we use the mean and standard deviation of the number of potentiations and depressions to assess the trial-to-trial variability in the case of same spikes and update seed. For the case of different spikes, we use the final late-phase weight value z_f as a test statistic for the strong protocols, and the maximum absolute deviation of the early-phase weight $|h-h_0|_{max}$ as a test statistic for the weak protocols. Although these statistics are skewed as will be demonstrated, a t-test of these statistics to compare the means would still be robust for finite variance and sufficient sample size (Chen, 1995). In this case, we consider that the sample mean is the one obtained from the emulation, and the true mean and the true variance are obtained from the simulation. The t-statistic becomes:

$$t = \frac{\bar{x} - \mu_0}{\sigma} \tag{3.40}$$

where $\bar{x}$ is the sample mean, μ_0 is the true mean, σ is the standard deviation. The null hypothesis is that the emulation mean is equal to the simulation mean. Using the test-statistic, we calculate the p-value which is a measure of the probability describing the likelihood of obtaining the observed data under the null hypothesis of a statistical test. We reject the null hypothesis at a p-value < 0.01, which indicates that there is a strong evidence against the null hypothesis. Despite not being able to prove that the means are equal, the p-value serves as a measure for evidence whether the simulation and emulation results are statistically significant.

Results

We present the results of our approach first on a single chip, specifically on emulating the calcium dynamics followed by the implementation of the plasticity rule. Then, we compare the final results of the implementation against different chips. Further results on other chips can be found in the appendix.

4.1 Emulating Calcium Dynamics

The adaptation parameters that are needed to emulate the calcium dynamics are tuned individually for each chip. Specifically, we focus on the calcium traces recorded using the MADC for judging the overall behavior compared to the theoretical calcium trace as well as the calcium mapping for judging the linearity of calcium around the potentiation and depression calcium thresholds.

4.1.1 MADC Calcium Traces

The retrieved calcium traces measured using the MADC at stimulation frequencies of 20 kHz and 100 kHz are presented in figures 4.1 and 4.2 respectively. At a frequency of 20 kHz, the calcium traces and the spikes align almost completely. At a frequency of 100 kHz, there is a misalignment between the theoretical and experimental calcium traces for two reasons. First, the adaptation circuit is non-linear beyond certain adaptation potentials. Second, there is a probability of either missing spikes or not recording all spikes at high frequencies. While a better mapping at 100 kHz can be achieved for lower adaptation reference potential and spike-triggered adaptation, this comes at the expense of noisy recordings at 20 kHz. Consequently, the calcium mapping should account for both, low and high frequencies.

4.1.2 CADC Calcium Samples

For the same frequencies and tuned adaptation parameters, the retrieved CADC calcium samples are shown in figures 4.3 and 4.4. Similar to the MADC results, there is a discrepancy between the theoretical and experimental calcium samples due to the frequency of the stimulation. Additionally, there exists a variability in the sampling times in the CADC, which means that the samples are not taken at exact multiples of the defined period. This is accounted for in figures 4.3 and 4.4 by error bars and is especially visible when the sampling times are around the spike

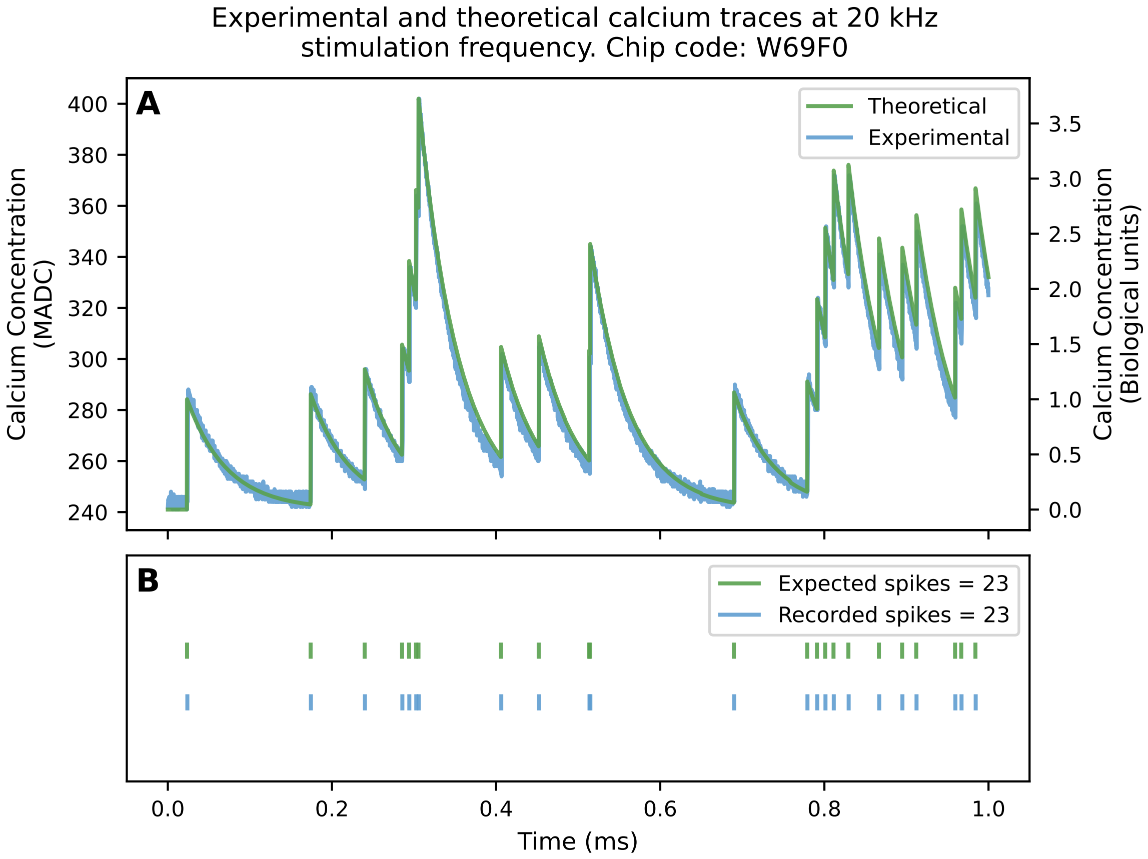


Figure 4.1: Emulation of calcium dynamics using MADC at 20 kHz stimulation frequency. (A) Theoretical calcium trace calculated from Poisson spikes using eq. (3.4) versus the experimental calcium trace extracted using the adaptation trace. (B) Expected Poisson spikes simulated at 20 kHz during 1 ms versus the recorded spikes from the parrot neuron that are used to generate the experimental calcium trace.

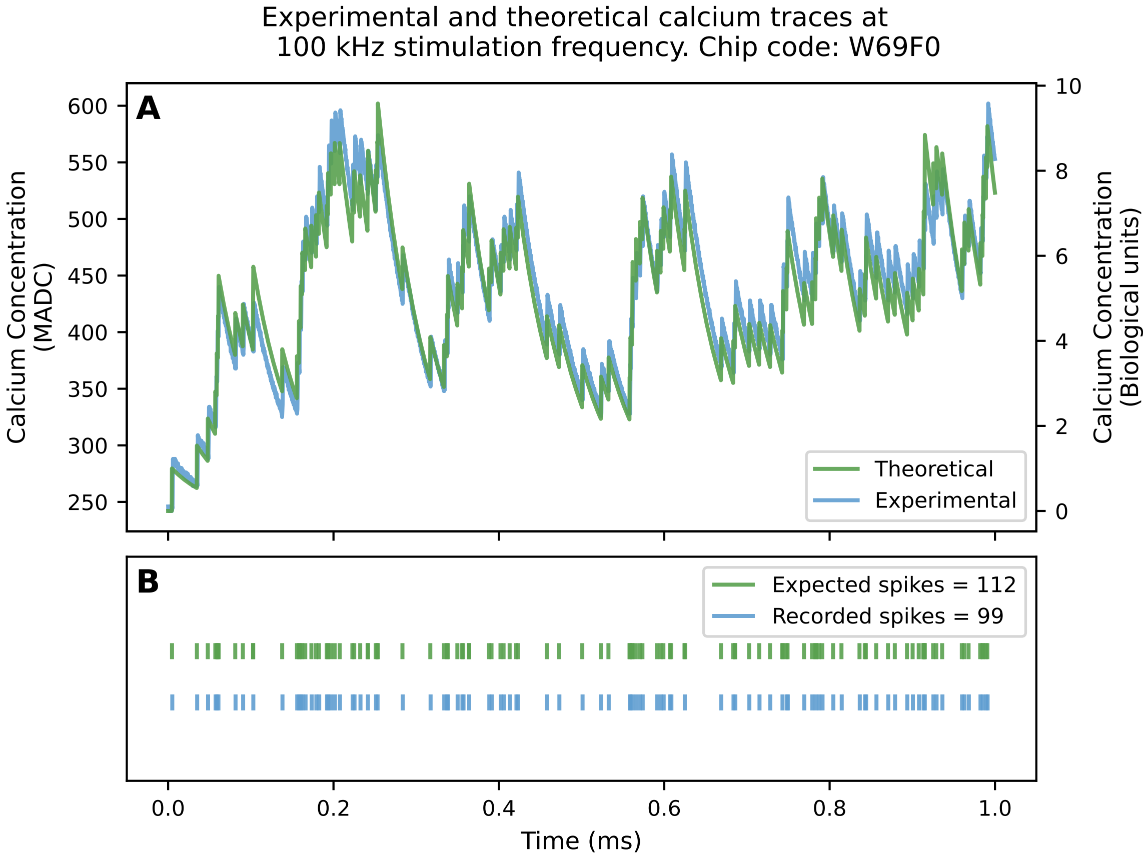


Figure 4.2: Emulation of calcium dynamics using MADC at 100 kHz stimulation frequency. (A) Theoretical calcium trace calculated from Poisson spikes using eq. (3.4) versus the experimental calcium trace extracted using the adaptation trace of the AdEx model from the recorded spikes. (B) Expected Poisson spikes simulated at 100 kHz during 1 ms versus the recorded spikes from the parrot neuron that are used to generate the experimental calcium trace.

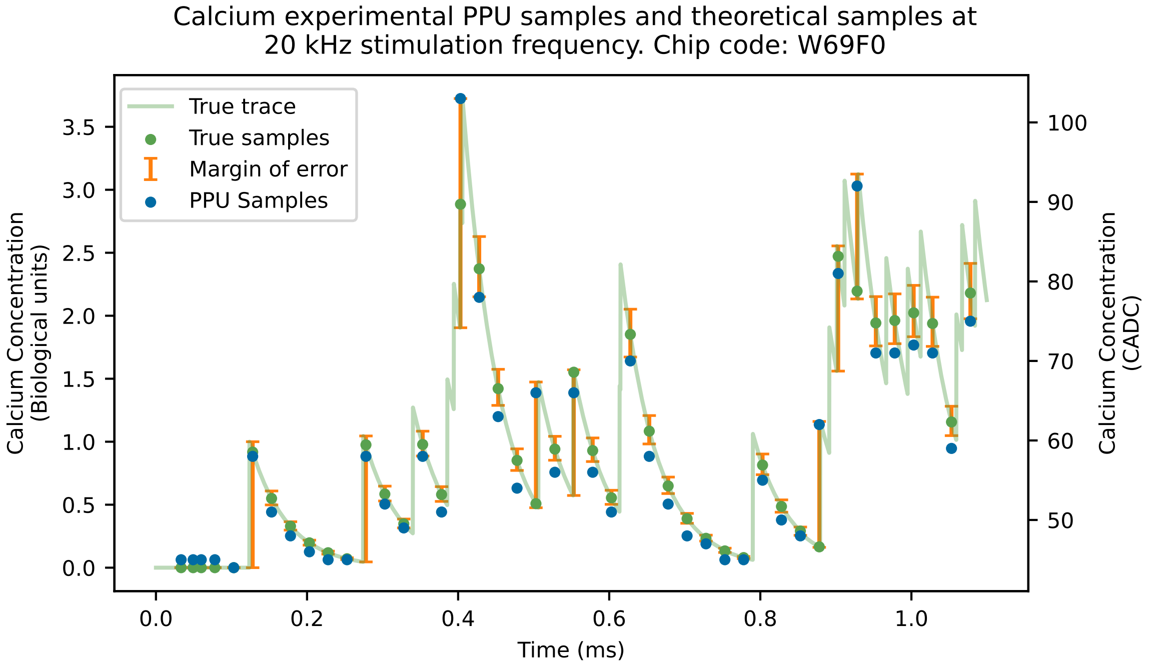


Figure 4.3: Emulation of calcium dynamics using CADC at 20 kHz stimulation frequency. The true trace is the theoretical calcium trace calculated using eq. (3.4) from the recorded spikes of the parrot neuron. The true samples are samples from the true trace at the times of the PPU samples. The PPU samples are generated by the CADC from the adaptation traces during the experiment. The margin of error shows the expected error at the PPU samples due to the variability in the sampling times.

times.

4.1.3 Mapping Calcium Thresholds

The results for mapping the biological potentiation and depression calcium thresholds θ_p and θ_d respectively are shown for a single chip in figure 4.5. The region under consideration until $c_{max} = 5$ is linear under the tuned adaptation parameters, and the linear regression is thus valid for mapping the calcium thresholds. The samples that are around the spike times are left-out as they are more affected by the variability in sampling times. These samples are shown along with their error bars in figure 4.5 to demonstrate how they can worsen the performance of the linear mapping. Further results for other chips can be found in the appendix.

4.2 Baseline Comparison

We reproduce the simulation results obtained using eq. (3.22) on the STC model equations. As mentioned earlier, these were described in details in the lab rotation (Atoui, 2024) , but are listed here again for comparison against the emulation results.

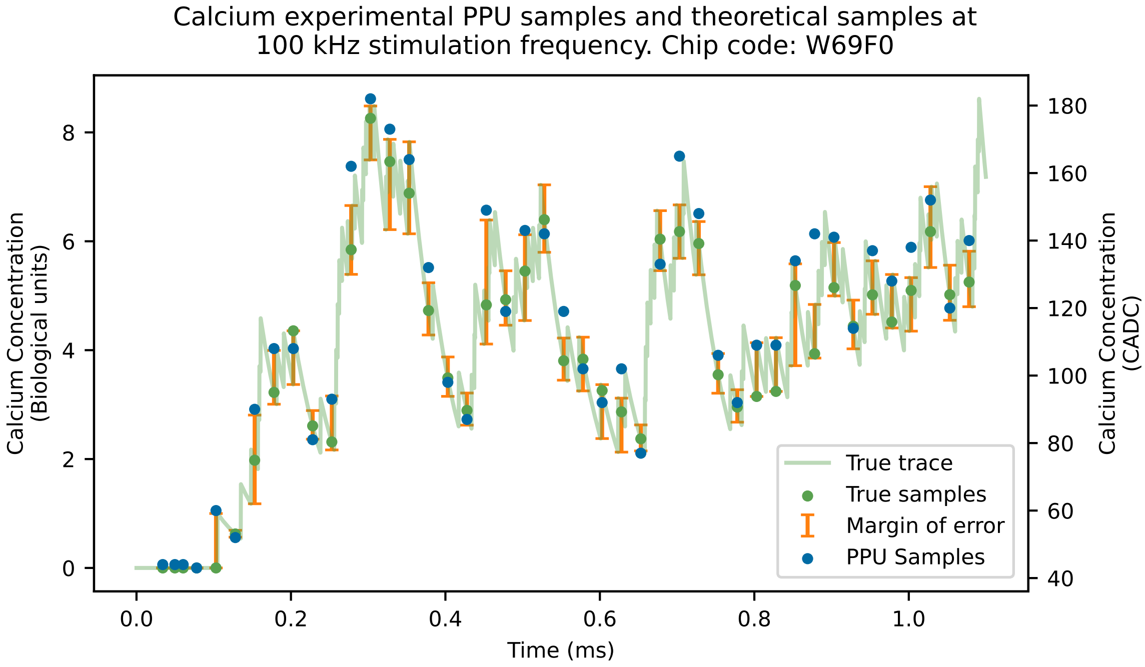


Figure 4.4: Emulation of calcium dynamics using CADC at 100 kHz stimulation frequency. The true trace is the theoretical calcium trace calculated using eq. (3.4) from the recorded spikes of the parrot neuron. The true samples are samples from the true trace at the times of the PPU samples. The PPU samples are generated by the CADC from the adaptation traces during the experiment. The margin of error shows the expected error at the PPU samples due to variability in the sampling times.

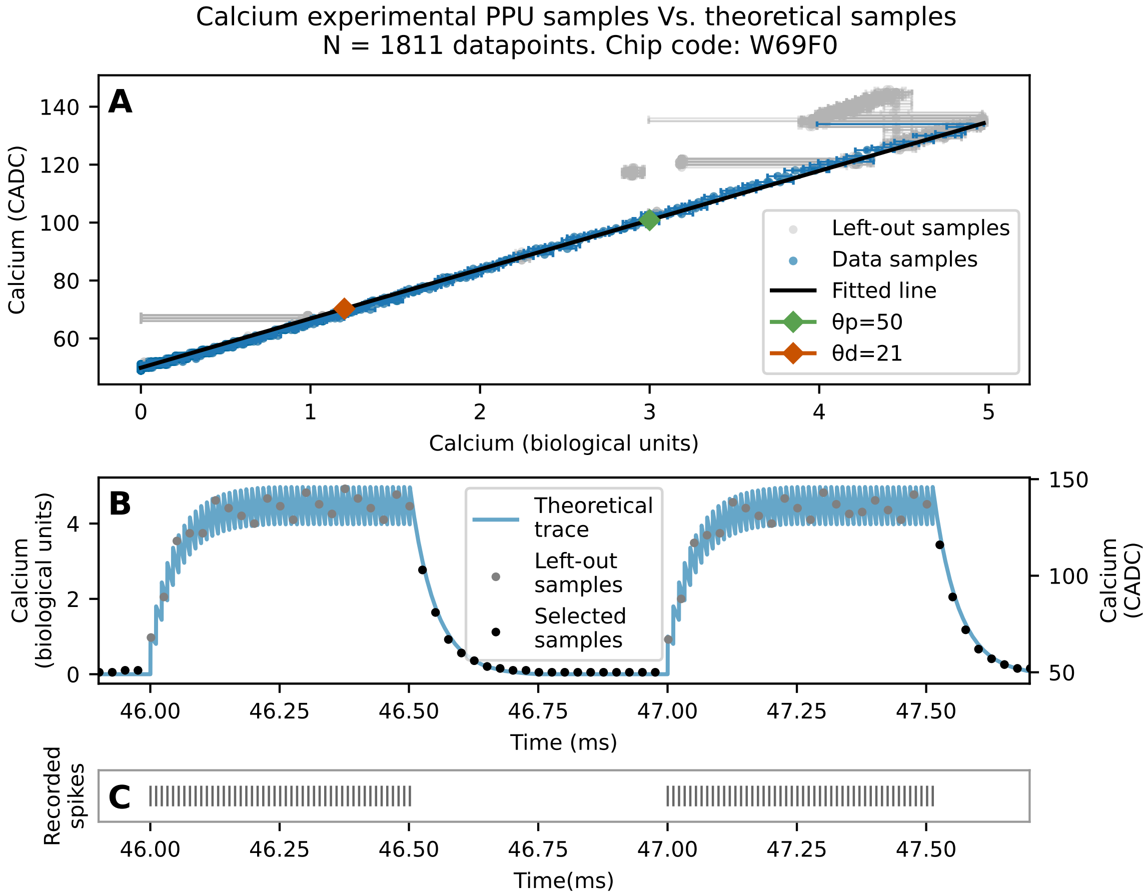


Figure 4.5: Mapping the calcium potentiation and depression thresholds using linear regression and least-squares estimation. (A) Calcium CADC values of the samples collected from running the calcium mapping experiment versus theoretical calcium calculated using eq. (3.4). The error bars are the expected errors from the CADC variability in sampling times. Samples with low expected errors are selected and fitted using least-squares method. The linear fit is used to estimate the potentiation and depression thresholds. (B) Theoretical calcium trace calculated from the recorded spikes of the parrot neuron using eq. (3.4) and PPU samples. The left-out samples are in the spiking duration of the mapping protocol whereas the selected samples are in the decaying region of the mapping protocol. (C) Recorded spikes of the parrot neuron used to generate the theoretical calcium trace and experimental calcium samples.

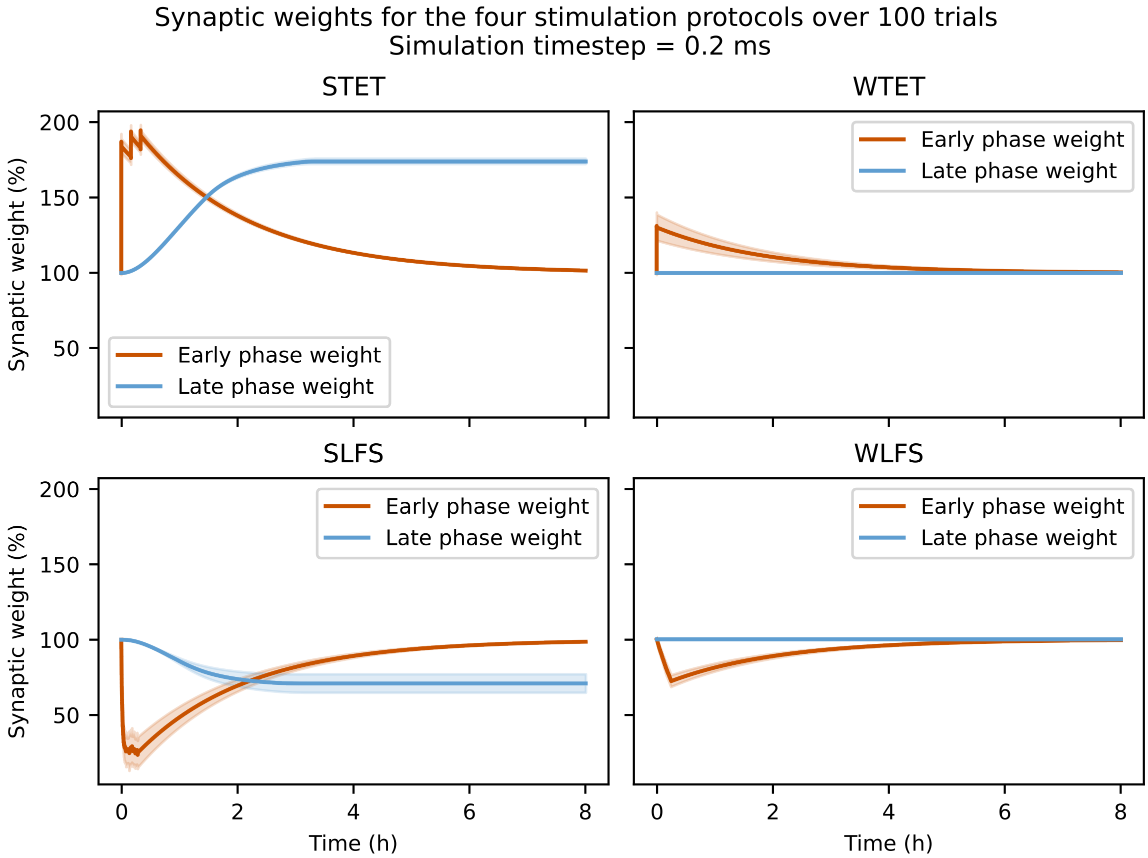


Figure 4.6: Impact of strong and weak tetanic and low-frequency stimulation protocols on a single synapse. The lines correspond to the average weights across the different spike trials, and the bands correspond to one standard deviation from the average. Tetanic protocols induce LTP while low-frequency protocols induce LTD. Strong protocols induce late-phase LTM while weak protocols induce only early-phase LTM. The simulation is carried out for 100 trials using the synapse and neuron parameters listed in table 3.1 and a simulation time step of 0.2 ms. The results are in agreement with those obtained in figure 2 of Luboeinski and Tetzlaff (2021)

4.2.1 Simulation at Base Timestep

The simulation at the base timestep of 0.2 ms is considered as the ground truth that we would ultimately like to achieve. The results of the simulation of the four protocols are presented in figure 4.6. These results are in alignment with the simulation results for a single synapse in work of Luboeinski and Tetzlaff (2021) except that we accounted for some hardware constraints for a fair comparison. The protein amount synthesized in each protocol is also presented in figure 4.7. Only the strong protocols can lead to protein synthesis and late-phase LTM. It is worth to note that there is an existing variability in the protein amount due to spikes for both STET and SLFS protocols.

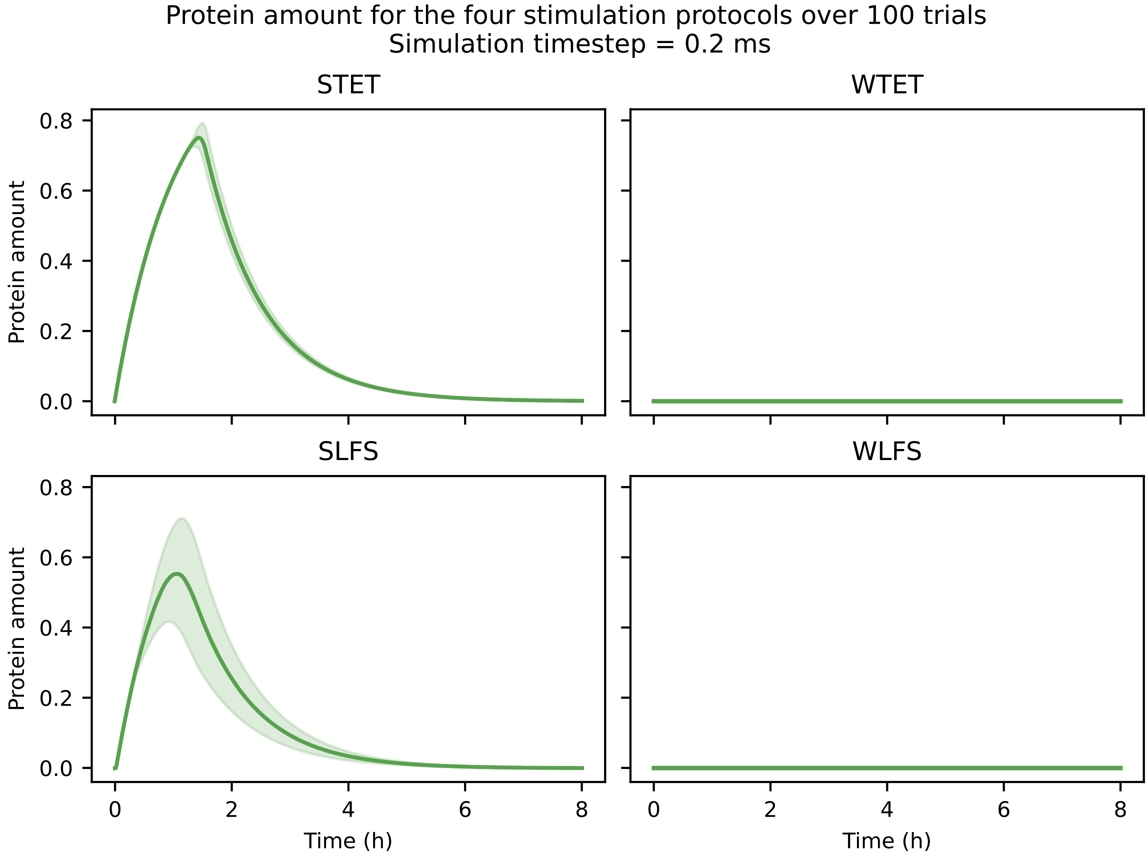


Figure 4.7: Impact of strong and weak tetanic and low-frequency stimulations on the synthesis of proteins. The simulation is carried out for 100 trials using the synapse and neuron parameters listed in table 3.1 and a simulation time step of 0.2 ms. The lines correspond to the average weights across the 100 trials, and the bands correspond to one standard deviation from the average.

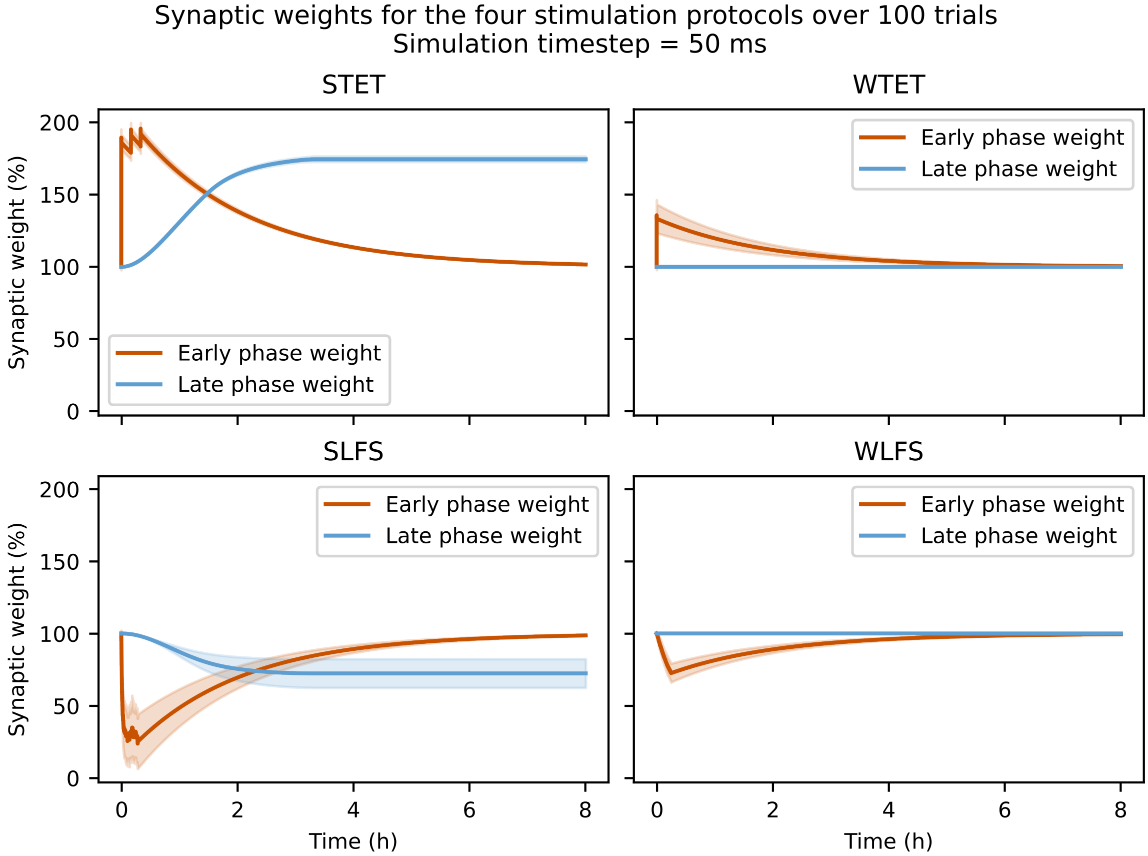


Figure 4.8: Impact of strong and weak tetanic and low-frequency stimulation protocols on a single synapse. The lines correspond to the average weights across the spike trials, and the bands correspond to one standard deviation from the average. Tetanic protocols induce LTP while low-frequency protocols induce LTD. Strong protocols induce late-phase LTM while weak protocols induce only early-phase LTM. The simulation is carried out for 100 trials using the synapse and neuron parameters listed in table 3.1 and a simulation time step of 50 ms. The results show some differences in the mean and standard deviation depending on the protocol with respect to those obtained at a simulation time step of 0.2 ms.

4.2.2 Simulation with Weight Updates at Hardware Timestep

To account for the major hardware constraint which is the update timestep, we repeat the simulation at a simulation timestep of 50 ms. This would serve as an immediate baseline to compare against the emulation results. The results of this simulation is presented in figure 4.8 for the four protocols. As an immediate comparison with the simulation at 0.2 ms, the variance in the early-phase weight and late-phase weight is higher in the SLFS protocol in the simulation at 50 ms as indicated by the error bands. The peak early-phase weight is also higher in the WTET protocol. The synthesized protein amount is also presented in figure 4.9 where the SLFS shows also a higher variance indicated by the error bands.

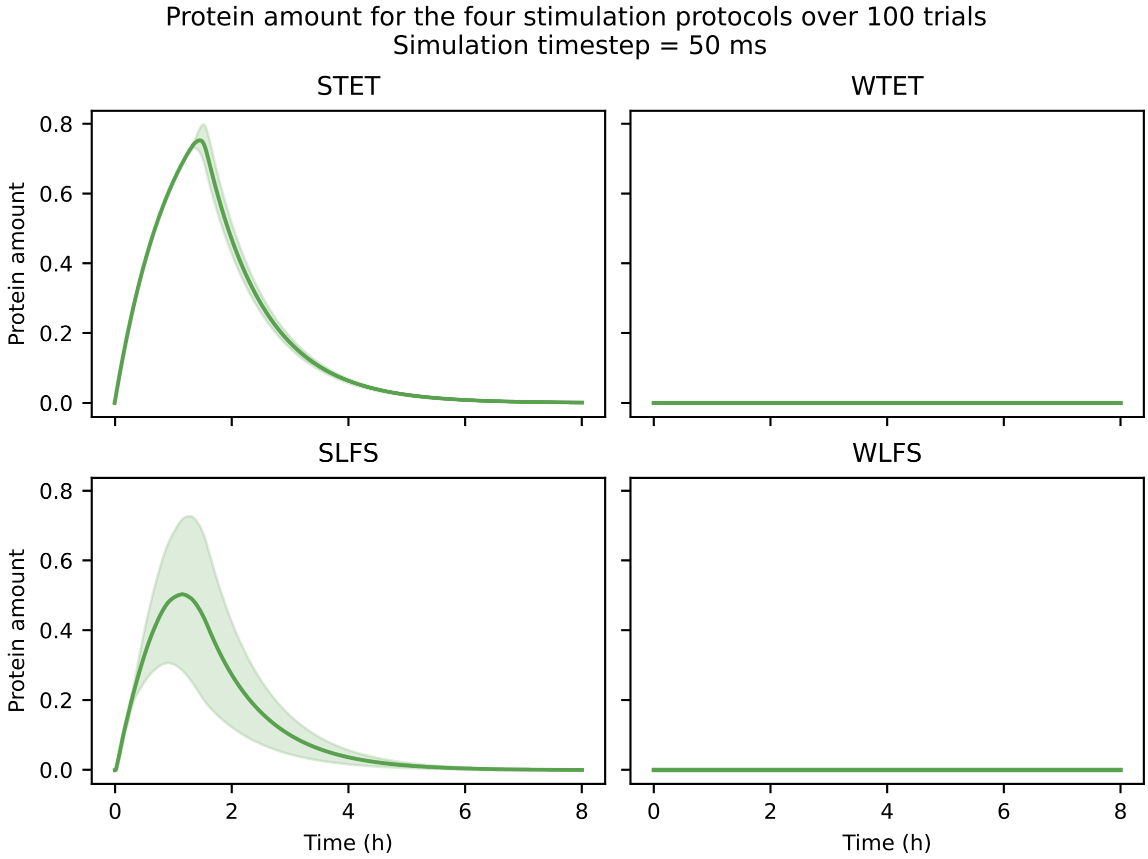


Figure 4.9: Impact of strong and weak tetanic and low-frequency stimulations on the synthesis of proteins. The simulation is carried out for 100 trials using the synapse and neuron parameters listed in table 3.1 and a simulation time step of 50 ms. The lines correspond to the average weights across the 100 trials, and the bands correspond to one standard deviation from the average. The results show some differences in the mean and standard deviation with respect to those obtained at a simulation time step of 0.2 ms.

4.3 Emulating a Single Synapse

In this section, we test our implementation of the STC plasticity rule on BrainScaleS-2. We start by showing with the baseline which is the single trial behavior that uses one set of spikes and one update seed. Then, we experiment with different update seeds and different spike trials.

4.3.1 Single Trial Behavior

A single trial of an experiment on BrainScaleS-2 uses one seed for simulating the presynaptic spikes and one seed for the stochastic rounding. As shown for figure 4.10, a single trial shows a clear discretization since the updates do not occur at every time step. The 100 trials performed in this section tests the reproducibility of the results. As we cannot judge on the performance of one trial and one chip, the trial-to-trial variability depends on the protocol, neurons, and spikes as will be explained later. For example, the results in figure 4.10 show a higher variability in the low-frequency protocols. However, this is not always the case as presented in the results found in figure A.2.1 in the appendix.

4.3.2 Average Behavior for One Set of Spikes

In this section, we run the experiment for 100 different update seeds using the same set of spikes. Each individual update seed yields different results but with a common overall behavior. We plot the individual seeds and the average behavior for the four protocols in figure 4.11. Averaging across the update seeds produces a smooth behavior, i.e. one that is obtained from an update time step of 0.05 ms. The traces no longer look discretized as in the single trial behavior. To investigate how close the average emulation behavior is to the baseline simulation behavior, we plot the same results for the same set of spikes in figure 4.12 for the four protocols. In this case, the emulation results agree with the simulation results for the STET and WLFS protocols but not for the SLFS and WTET protocols. While these results are not to be generalized across different spike trials, they can be explained by the differences between the number of potentiations and depressions between the simulation and emulation shown in figures 4.13. The number of potentiations and depressions is almost identical in the STET and WLFS protocols. For the SLFS protocol, the timing differences lead to differences in the evolution of the synaptic weights. The number of potentiations and depressions are significantly different in the SLFS protocol and WTET protocols which cause the difference in the synaptic weights. All these differences are primarily attributed to the variability in the sampling times.

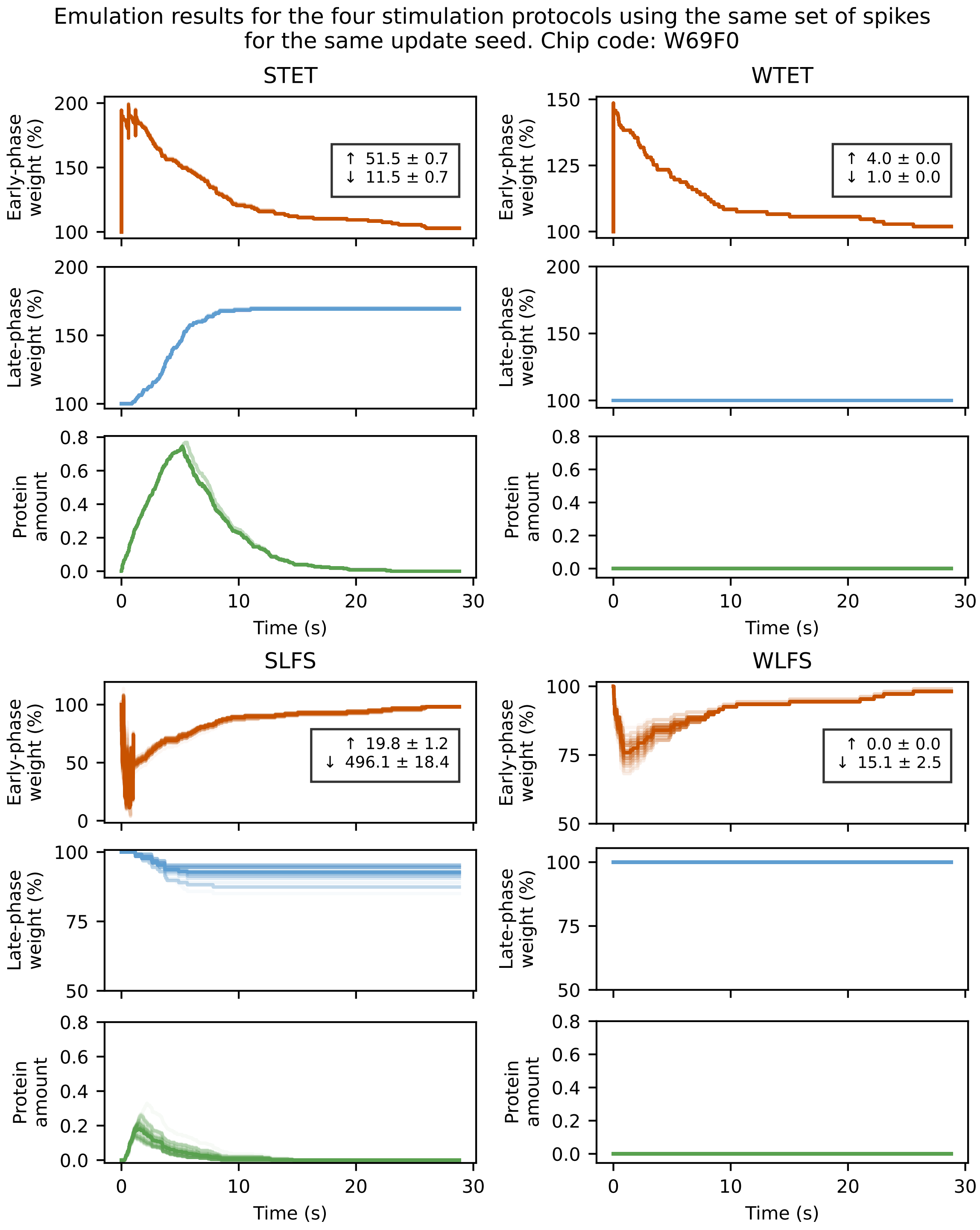


Figure 4.10: Emulation results for the four stimulation protocols for one set of spikes repeated 100 times for a single update seed. The aim is to check the reproducibility of the results over different trials. The source of variability is the adaptation circuit. The measures of performance are the mean and standard deviation of the number of potentiations and depressions shown by arrows pointing upwards and downwards respectively for each protocol.

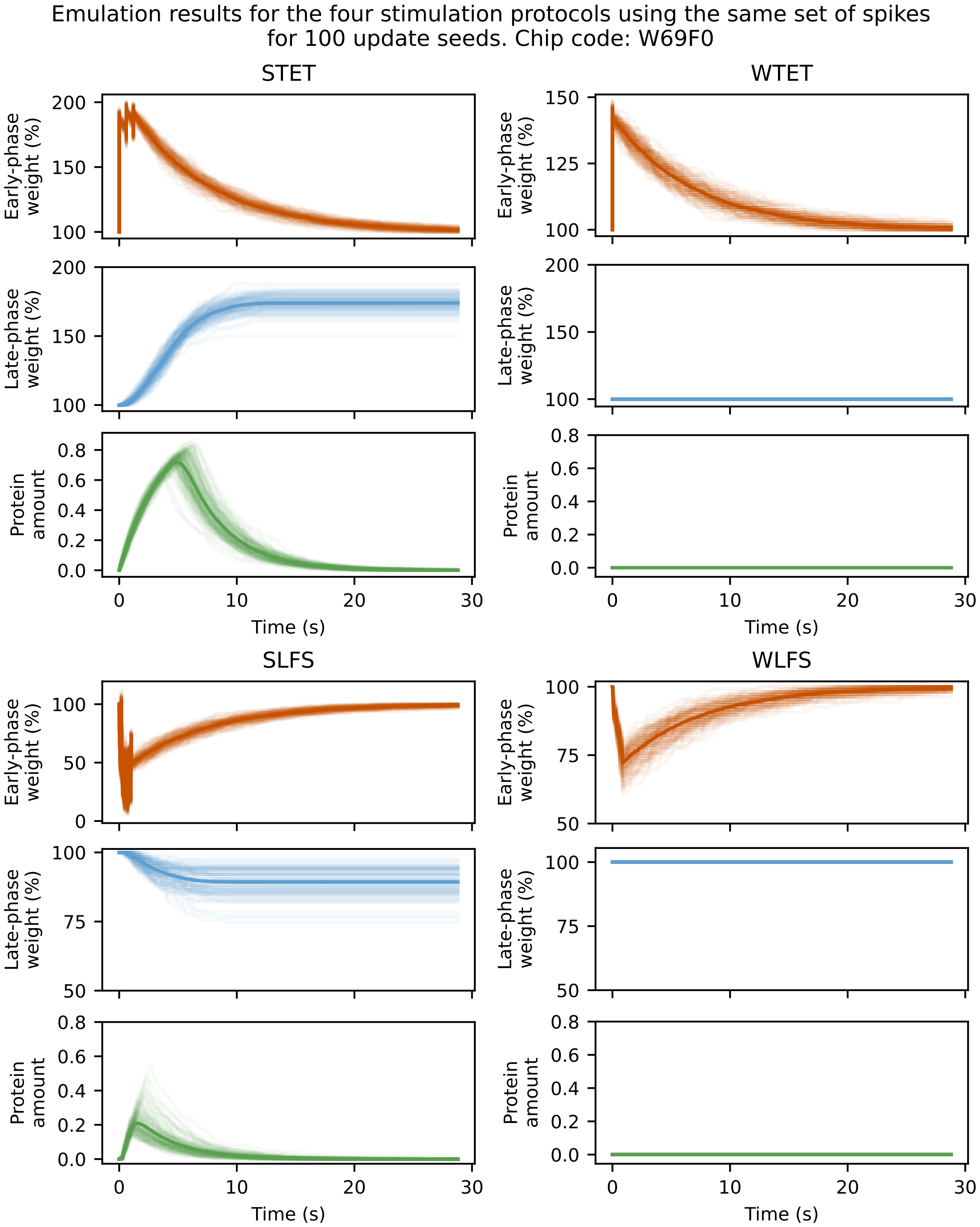


Figure 4.11: Emulation results for the four stimulation protocols using one set of spikes and different update seeds. The trajectories in faded color show the individual trials, and the trajectory in bold shows the average behavior. The overall average behavior is in agreement with the behavior obtained by simulation for the four protocols.

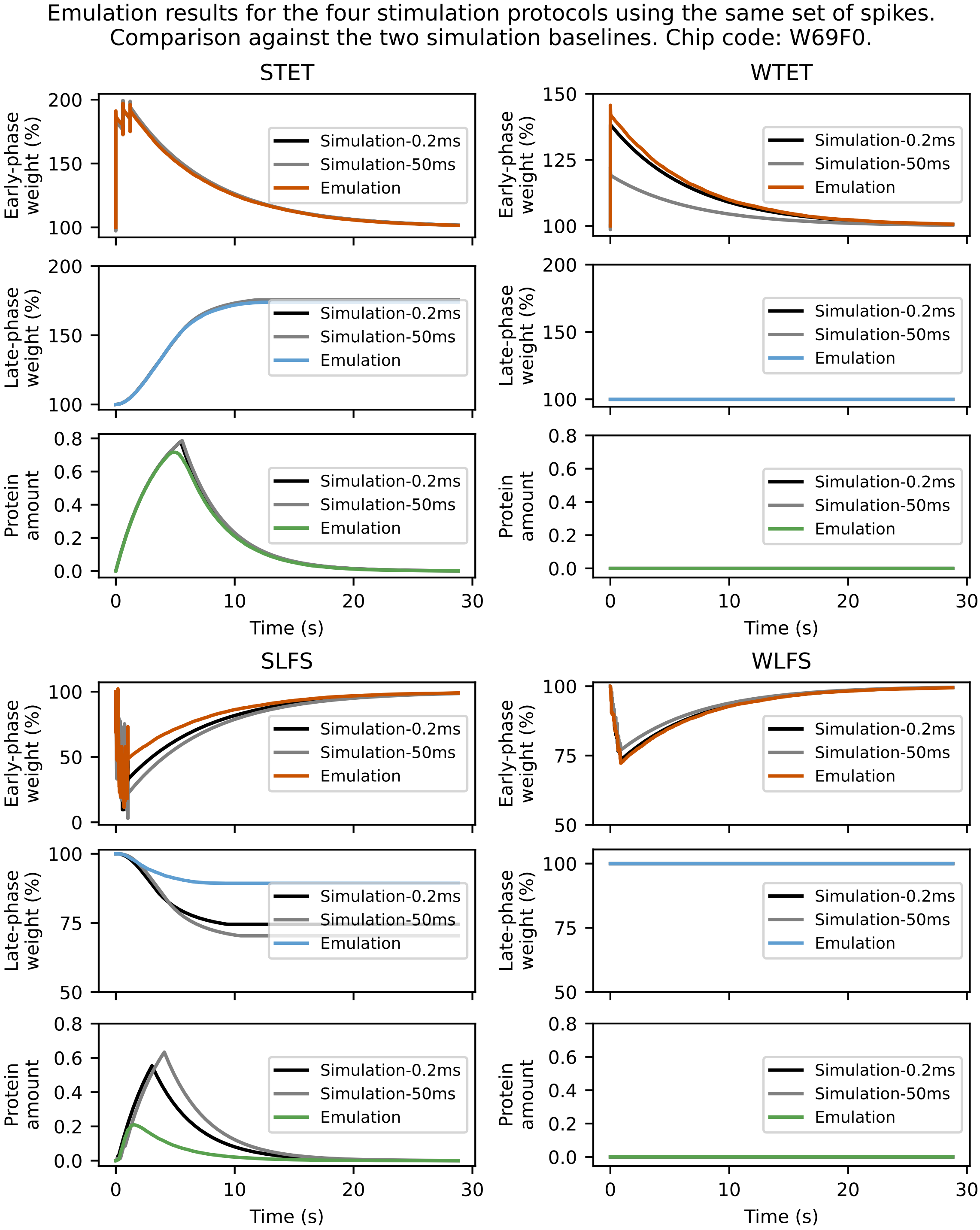


Figure 4.12: Comparison of the emulation results of the four stimulation protocols against the simulation baselines for 100 update seeds. The agreement between the emulation and simulation results depends on the stimulation protocol. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

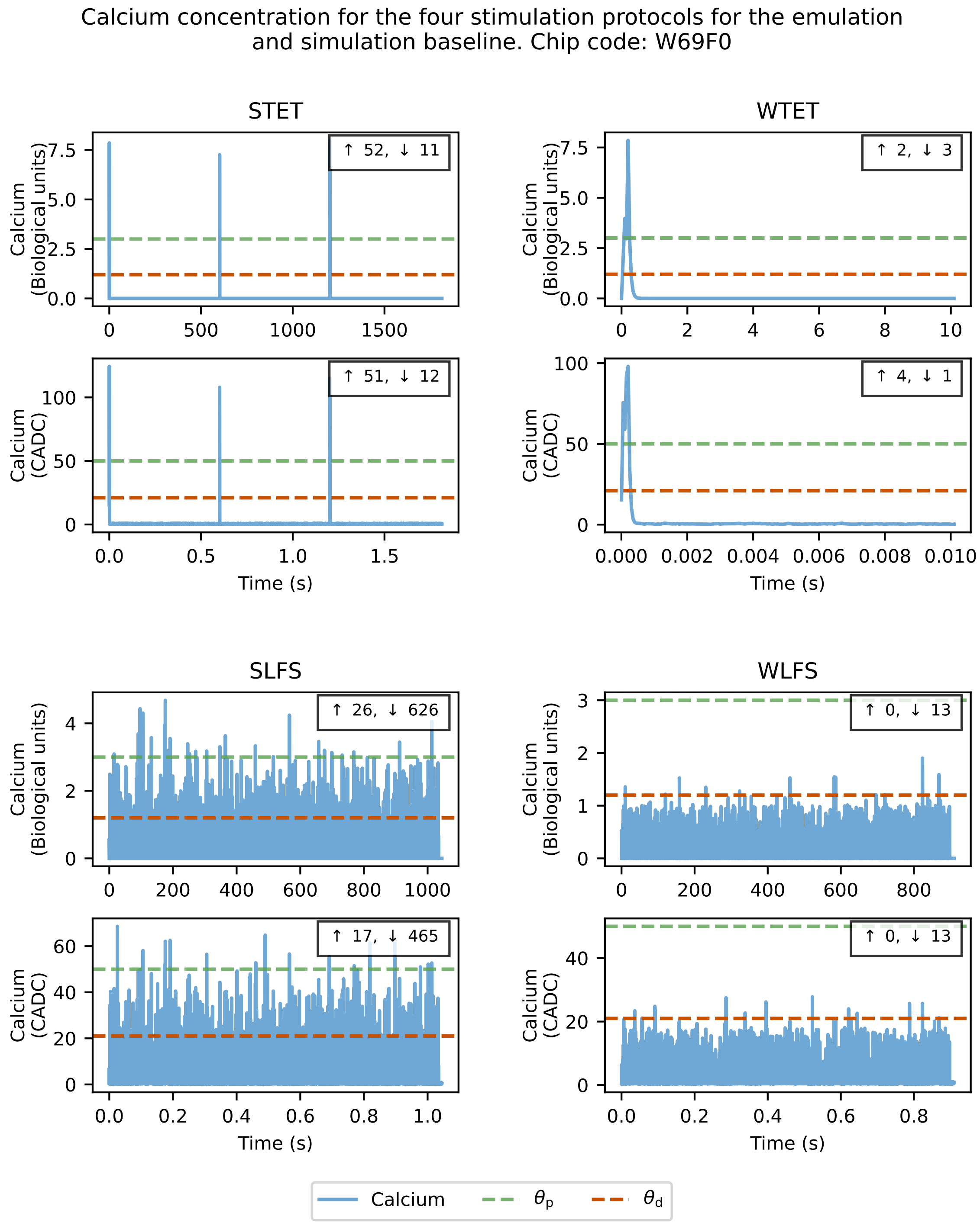


Figure 4.13: Comparison between the calcium concentration in the emulation and simulation schemes for a single set of spikes in the four stimulation protocols. The average number of potentiations and depressions for each protocol in the simulation and emulation cases are indicated by the numbers on the upper right of each figure by the up and down arrows respectively. The small difference in the number of potentiations and depressions between the simulation and emulation explains the alignment in the traces of the STET and WLFS protocols. The larger difference in the number of potentiations and depressions between the simulation and emulation explains the difference in the traces of the WTET and SLFS protocols.

4.3.3 Average Behavior across Different Sets of Spikes

In this section, we experiment with 100 different trials of spikes. We plot the results against the two simulation baselines and use the final late-phase weight and maximum absolute deviation of the early-phase weight as test statistics for the strong and weak protocols respectively.

Comparison with Simulation at Base Time Step

Figure 4.14 show the results of the emulation plotted against the simulation performed at a time step of 0.2 ms. The overall behavior of the protocols aligns with the ground truth simulation at 0.2 ms. There are differences in the mean of the synaptic weights and protein amount in all protocols. However, there are significant differences in the variance shown by the bands corresponding to the standard deviation.

Comparison with Simulation at Hardware Timestep

The same results are plotted against the simulation baseline performed at a time step of 50 ms for a fair comparison in figure 4.15. The mean of the synaptic weights and protein amount seem to align with those obtained from the simulation with small differences in the mean. The variances of the emulation seem close to the simulation for all protocols except for the STET protocol. This shows that the effects in the hardware variability that were demonstrated for one set of spikes also seem to cancel out.

4.3.4 Comparison across Different Chips

We perform the experiments for the different spikes for two other chips, and we compare the results against the 0.05 ms. Table 4.1 show the tuning parameters for the adaptation circuit used in the four chips we used. Figures 4.16, 4.17 and 4.18 show the results for the four protocols in three different chips. The average behavior of the protocols align with the expected behavior. For all chips, the STET protocol aligns with the simulation baseline. The main differences in the mean lie in the low-frequency protocols. Furthermore, the variance of the emulation aligns with the variance of the simulation at the 0.05 ms timestep, indicating that the hardware variability effects cancel out.

Finally, we compare the results across the chips numerically using the mean, standard deviation, and p-value of the test statistics which are the final late-phase weight z_f in the strong protocols and the maximum absolute deviation $|h - h_0|_{max}$ of the

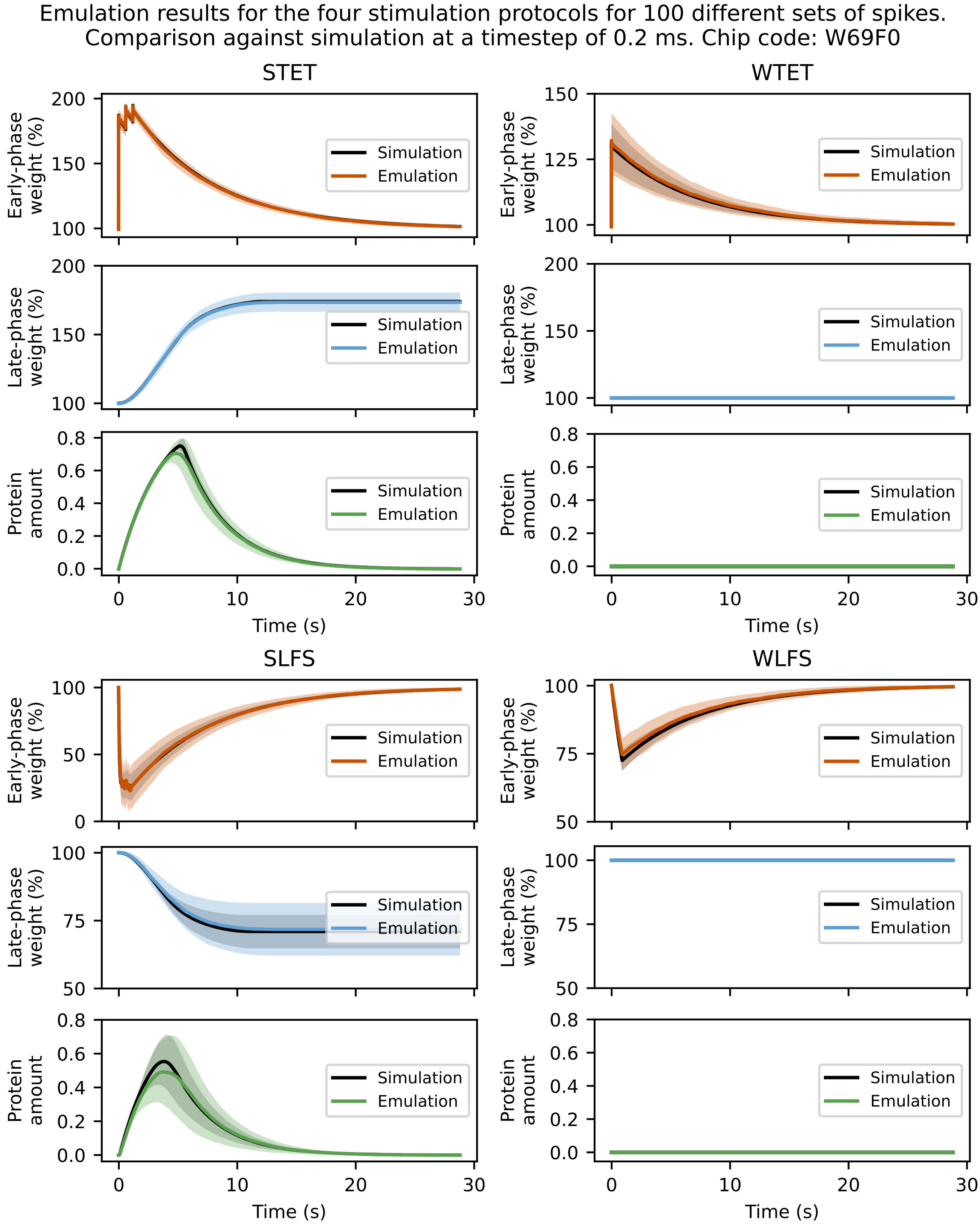


Figure 4.14: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 0.2 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

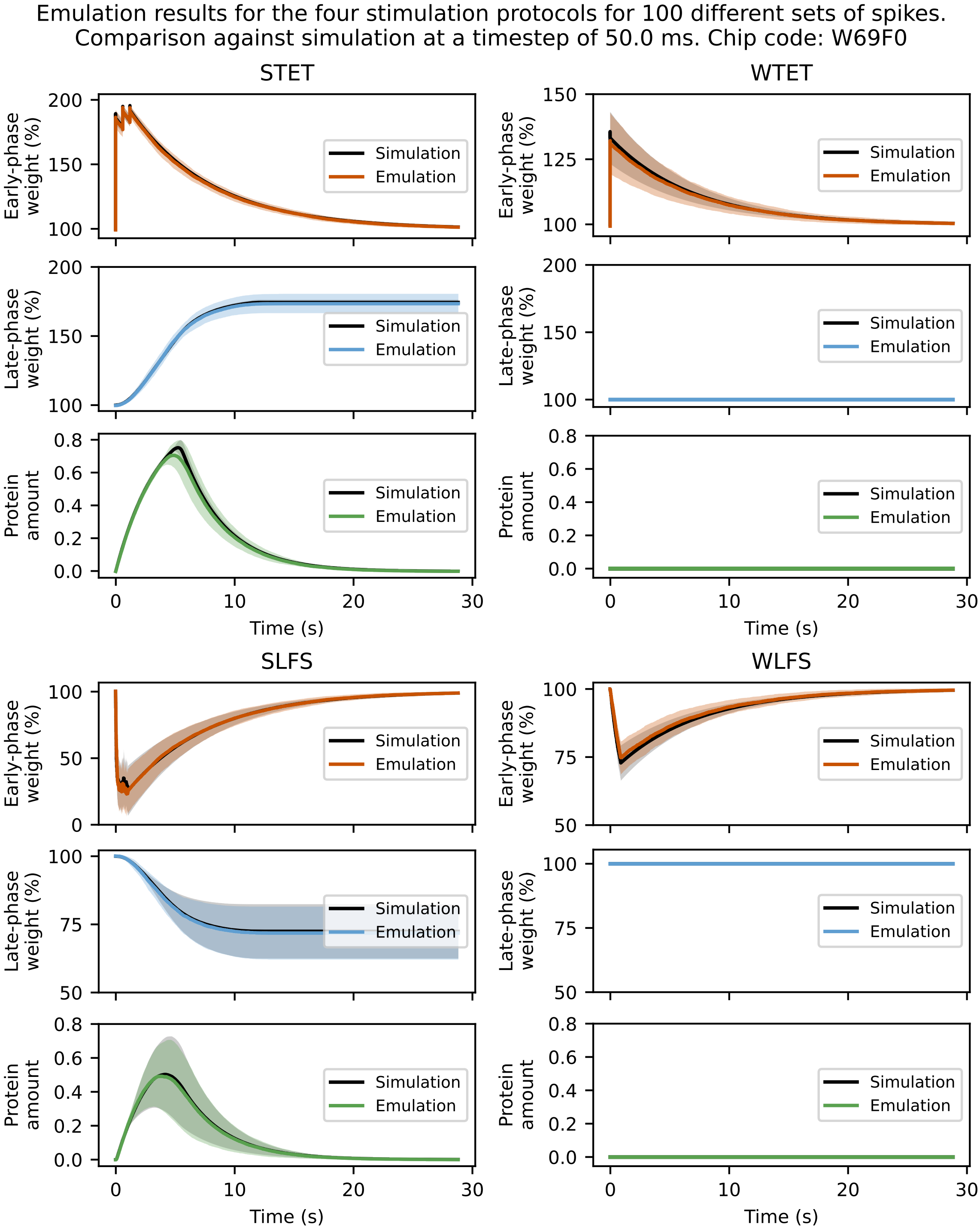


Figure 4.15: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 50 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

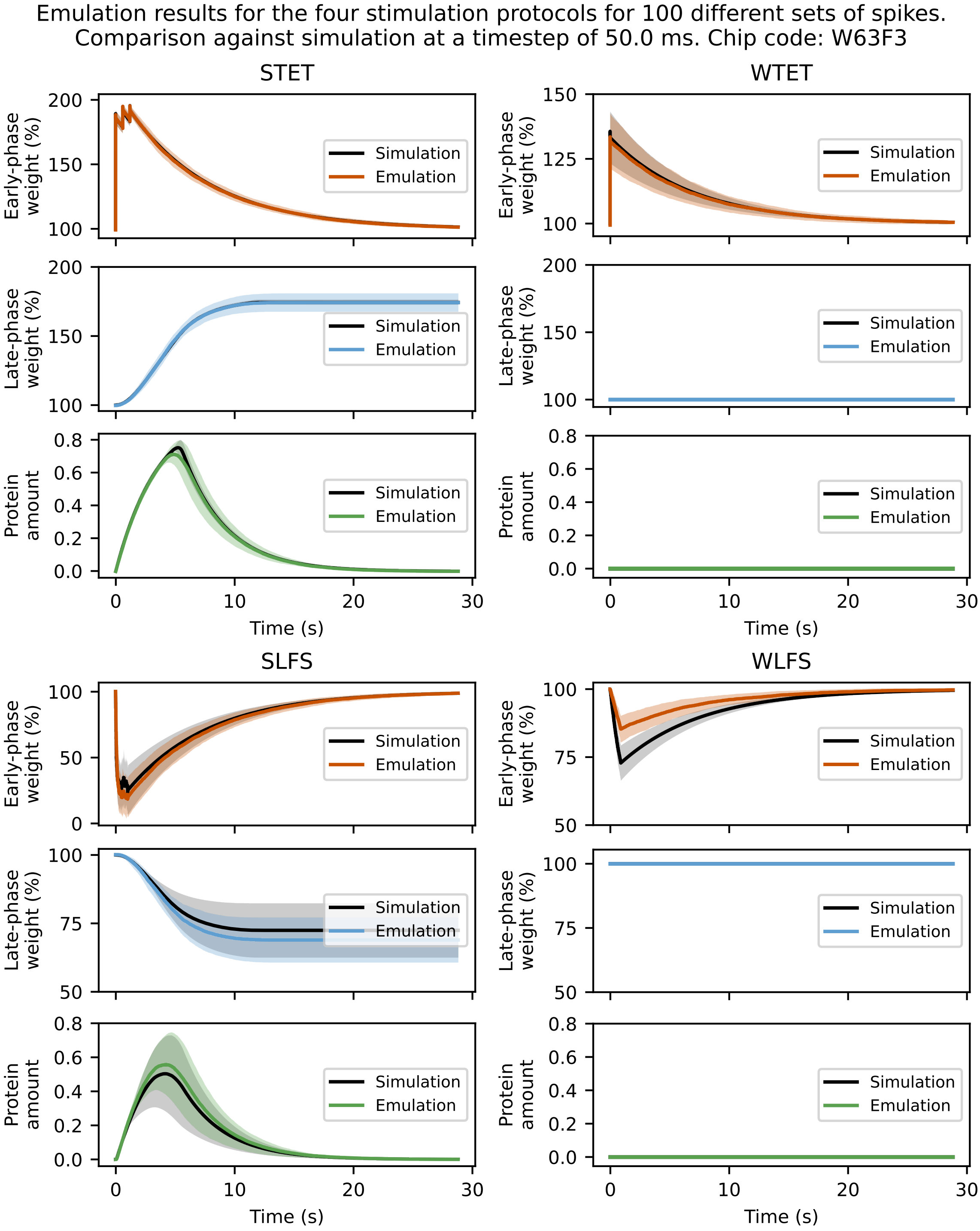


Figure 4.16: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 50 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

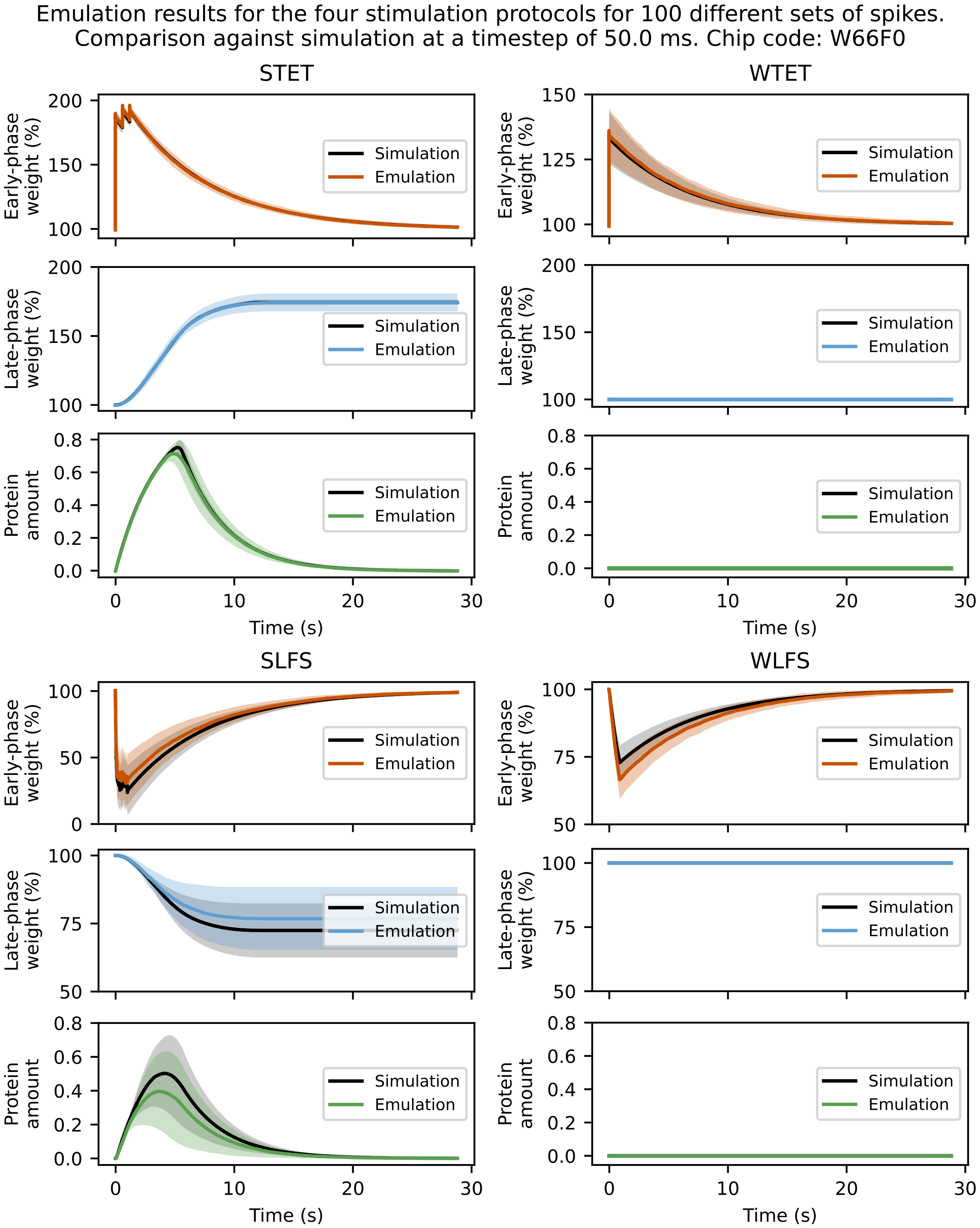


Figure 4.17: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 50 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

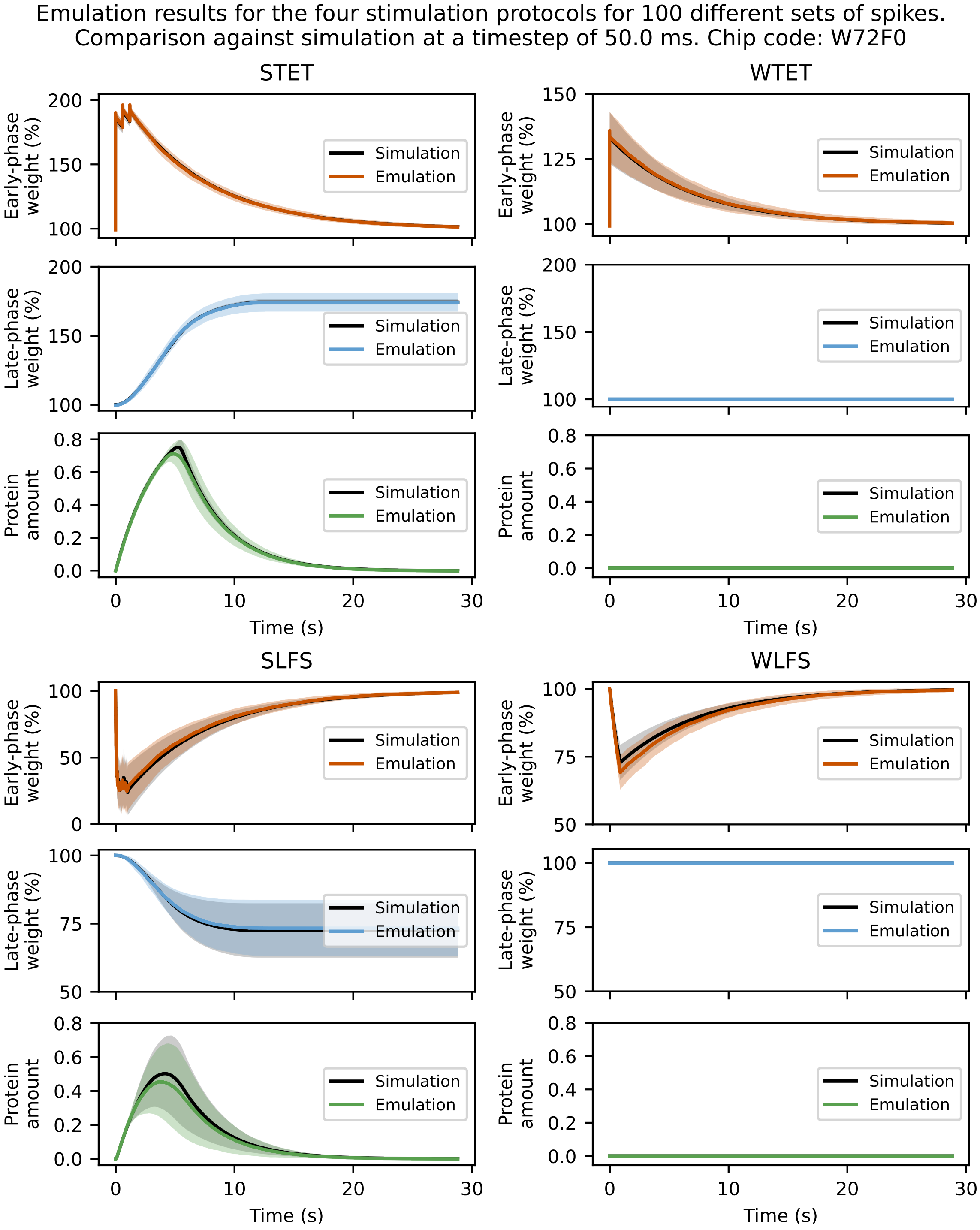


Figure 4.18: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 50 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

Chip	Adaptation Reference Potential	Adaptation Time Constant	Spike-triggered Adaptation (b)
63F3	440	280	400
66F0	400	290	460
69F0	400	320	560
72F0	450	320	620

Table 4.1: Parameter values used for tuning the adaptation circuit in different chips

early-phase weight in the weak protocols. Table 4.2 shows the mean and standard deviation of the test statistics for the four protocols in the two simulation schemes. As shown, there are differences that arise in the mean and standard deviation due to increasing the simulation timestep. Table 4.3 shows the mean, standard deviation, and p-value of the test statistics for the four chips we experimented with for the four stimulation protocols. The differences in the mean and standard deviation demonstrate the variability between the chips. The mean of the test statistics is similar between the emulation and simulation for all protocols, but the variance is higher for the STET protocol in the emulation case compared to other protocols. The null hypothesis is not rejected for all protocols and chips except for one p-value for the WLFS protocol in chip W63F3 which showed statistical significance.

Protocol	Simulation Timestep	Mean	Standard deviation
STET	0.2 ms	0.739	0.018
	50 ms	0.743	0.019
WTET	0.2 ms	0.132	0.037
	50 ms	0.152	0.039
SLFS	0.2 ms	-0.290	0.0616
	50 ms	-0.275	0.0997
WLFS	0.2 ms	0.114	0.0165
	50 ms	0.115	0.0272

Table 4.2: Mean and standard deviation of the test statistics for the four stimulation protocols in the simulation scheme for the two simulation timesteps, 0.2 ms and 50 ms. The test statistics are the final late-phase weight for the strong protocols and the maximum absolute deviation of the early-phase weight in the weak protocols.

Protocol	Chip	Mean	Standard deviation	p-value (0.2 ms)	p-value (50 ms)
STET	W63F3	0.739	0.067	0.962	0.860
	W66F0	0.742	0.065	0.872	0.945
	W69F0	0.734	0.0702	0.805	0.652
	W72F0	0.739	0.067	0.958	0.863
WTET	W63F3	0.145	0.0408	0.726	0.855
	W66F0	0.156	0.0379	0.519	0.926
	W69F0	0.141	0.0448	0.806	0.779
	W72F0	0.155	0.0377	0.535	0.944
SLFS	W63F3	−0.310	0.0825	0.7408	0.726
	W66F0	−0.231	0.115	0.345	0.661
	W69F0	−0.281	0.0969	0.892	0.949
	W72F0	−0.265	0.103	0.695	0.923
WLFS	W63F3	0.062	0.0205	0.0018*	0.056
	W66F0	0.141	0.0286	0.115	0.327
	W69F0	0.106	0.0253	0.591	0.756
	W72F0	0.129	0.026	0.388	0.586

Table 4.3: Evaluation of the emulation results across four different chips using the test statistics of the four stimulation protocols. The test statistics used are the final late-phase weight for the strong protocols and the maximum absolute deviation of the early-phase weight in the weak protocols. The p-value is calculated using the mean of the test statistic of the emulation and using the mean and variance of the same statistic in the simulation scheme as true values. Results with (*) show statistical significance.

Discussion

In this work, we emulated the synaptic tagging and capture (STC) plasticity rule for a single synapse on the mixed signal neuromorphic platform BrainScaleS-2. We used the work of Luboeinski and Tetzlaff (2021) to emulate the equations of the plasticity rule. The neuron and calcium dynamics were emulated by the analog core of BrainScaleS-2, while the plasticity equations were provided to the plasticity processing unit (PPU) for weight updates. We defined the hardware constraints, specifically the time updates, finite arithmetic, and hardware variability. Then, we proposed to overcome these constraints through increasing the update timestep, stochastic rounding, and tuning of the adaptation circuits for emulation of calcium dynamics. We used the standard plasticity protocols to assess our approach as well as general performance measures such as the mean, standard deviation, and p-value. We also used statistics that are specific to the plasticity rule, namely the number of potentiations and depressions, the final late-phase weight in the strong protocols, and the maximum deviation of the early-phase weight in the weak protocols.

Our approach faithfully reproduces the simulation results of the single synapse for four stimulation protocols with small differences in the mean and standard deviation across protocols. This highlights the power of stochastic rounding for finite arithmetic and reduced precision. The differences between the emulation at a time update of 0.05 ms and the simulation at a time update of 0.05 s can be primarily attributed to the hardware variability in the sampling times which affect the results in case the calcium is at the edge of potentiation, depression, or at a rising edge of the spike. This is sensitive because the calcium dynamics are fast and can decay quickly or rise almost instantaneously by the effect of the spike. Another important consideration is tuning the operation range of the adaptation circuit that is used to emulate the calcium dynamics. The tuning has to account for circuit linearity and the operation at all considered stimulation frequencies. This presents a challenge since there is a trade-off between the three objectives: Operation at low frequencies requires a high spike-triggered adaptation (b value) so as not to obtain a small depression threshold that can be easily impacted by noise or decay. Operation at high frequencies requires a low spike-triggered adaptation to maintain circuit linearity. In both cases, the reference potential and the adaptation time constant have to be tuned accordingly. In this work, tuning was assessed visually for linearity and alignment of calcium traces with their biological duals at high and low frequency, but this should be optimized using further numerical tests and several iterations as done for calibrating the neuron circuits in the work of Weis (2020).

Our experiments on BrainScaleS-2 feature different sources of variability. Trial-to-trial variability appears when running the experiment using the same spikes and update seeds for several trials. Using different update seeds for the same spikes also introduces a form of variability. There also exists the variability that arises from using different spike trials, which is the only type of variability found in the simulation scheme. Finally, there exists a variability between different chips that arises from the manufacturing process. The trial-to-trial variability specifically arises from the analog emulation of calcium which, due to its analog nature, is not fully reproducible between trials. Furthermore, the emulation of calcium differs across protocols and is also dependent on the calcium tuning and spikes under consideration. For example, spikes with lower interspike interval (ISI) tend to provide more robust results compared to spikes with higher ISI since they would exceed the threshold by a larger amount and can overcome the effect of the variability in the sampling times. This effect of ISI introduces a variability across trials, but also appears significantly in the comparison of the average emulation results against the simulation baseline for one set of spikes. In general, having a high calcium amount leads to more reproducible results compared to low calcium amounts, and this appears in the high variance of depressions in the low-frequency protocols compared to the low variance of potentations in the tetanic protocols for the trial-to-trial variability experiment. For the random update seeds we considered in the same set of spikes, the different trials showed variability, some of which can be considered as extreme. This should be further investigated in terms of choosing the pseudo-random number generator and seeds, especially that we are considering only 100 trials.

Our implementation of STC on BrainScaleS-2 would be considered feasible if the overall variability stems from the spike variability, as in the simulation case. In other words, the errors that arise from the variability of the hardware would be compensated across different trials. We expected that the effects of the variability of sampling times and the ISI would cancel across different set of spikes; in the same way that the variability in sampling time would lead to underestimation of potentiations or depressions for some sets of spikes, it could also lead to overestimation of potentiations or depressions for other sets of spikes. This appears true in the experiments across different sets of spikes, where the variance of the test statistics in the emulation is close to the variance of the test statistics of the simulation at an update time of 0.05 s for all protocols except for the strong tetanic stimulation (STET) protocol where the variance seems relatively higher. This shows that for experiments across different sets of spikes, the main source of the variability is the spikes, and variability that stems from analog computing seems relatively negligible which makes our implementation successful.

For the STET protocol, the hardware variability is more apparent than the spike variability as this protocol is considered robust due to its high calcium concentration. This makes it less prone to the variability of the sampling times. Since this protocol is important for the network emulation, we can attempt to resolve this problem by running more experiments for the STET protocol. Finally, the chip variability was depicted by tuning the adaptation circuits used for emulating the calcium dynamics, which gave rise to different potentiation and depression thresholds and different performance across protocols. This demonstrates the importance of automating the tuning process to obtain reasonably close results between different chips.

As far as the emulation of the network in the work of Luboeinski and Tetzlaff (2021) is concerned, using BrainScaleS-2 for network experiments is feasible though with further challenges. In this network, there are excitatory neurons with strong recurrent connections that are referred to as a cell assembly in addition to excitatory control neurons. Within the cell assembly, a portion of the neurons are stimulated with a learning protocol, and a sub-portion with a recall protocol. These protocols resemble the STET protocol that we investigated, which proved to be robust with the least variability and closest average behavior among other protocols, even when tested on different chips and a slower update time. The reason is that the STET protocol is more immune to the variability in sampling times compared to other protocols as the calcium concentration is high compared to the potentiation threshold due to the high number of spikes. However, the tuning of the adaptation parameters is still important for the calcium concentration of the neurons that are not stimulated in the cell assembly or are outside the cell assembly as these neurons are affected by a lower number of spikes and thus possess a moderate amount of calcium. Another form of variability that will be introduced in the network emulation is the variability in the neuron circuits involved in the emulation of the neuron dynamics. For a single synapse, there were almost no postsynaptic spikes, and this is the case in the emulation and simulation schemes. The network experiment requires tuning the neuron circuits and the synaptic weight to resemble the biological network. One challenge is that the synaptic weights have a low resolution (0-63 least significant bits (LSB)) compared to the calculated early-phase and late-phase weights, but we expect that this would also be compensated across different weights. Another challenge would be the update of synapses in the network that will not occur at the same time; here, we will also experiment with stochastic updates which introduce another form of variability. Finally, we would have to connect different chips to achieve the desired number of neurons in the network.

An important advantage of the use of BrainScaleS-2 in this emulation is the significant speed-up compared to the simulation performed on standard computers. For example, one challenge faced by Luboeinski and Tetzlaff (2021) in the network simulation is the long simulation runtime and the need to perform some approximations to reduce this runtime. Our implementation features a significant speedup with an average runtime of 40 s per trial for the 8-hour biological-time experiment. In a similar fashion, we expect to reduce this runtime in the network emulation which allows running experiments for different network topologies and a higher number of spike trials. On the other hand, our implementation possesses hardware limitations in which the axonal spike delay, postsynaptic calcium influx delay due to presynaptic spikes, and calcium noise fluctuations during potentiation and depression were not implemented. The effect of these limitations is not visible in the emulation of a single synapse due to the major absence of postsynaptic activity, and this must be further investigated in the network behavior. However, the use of analog hardware for the emulation of the neuron and calcium dynamics intrinsically presents delays and some forms of variability. The question whether analog computing is advantageous in this case and can provide realistic dynamics is of significant interest.

Besides being used for emulating biological neural networks to understand the human nervous system and behavior, the STC plasticity rule could be used as a learning rule for machine learning tasks, similar to spike-timing dependent plasticity (STDP). We get the inspiration from the biological plausibility of calcium-based plasticity and its relevance to many behavioral experiments. A challenge for this application is the relative computational steps needed to implement this plasticity rule compared to other plasticity rules. This necessitates simplifying the rule or implementing it along other algorithms that can exploit the key advantages of STC.

Conclusion

The STC plasticity rule can be faithfully emulated on the neuromorphic circuits of BrainScaleS-2. The hardware constraints related to the update timestep and reduced precision are overcome by using a higher timestep for updating the synaptic weights and stochastic rounding. The emulation of the STC plasticity rule requires a comprehensive procedure for emulating the neuron and calcium dynamics and mapping the biological parameters to hardware. The use of analog hardware produces different sources of variability that lead to small errors in the mean of our test statistics, and the effect of these forms of variability depend on the stimulation. Nevertheless, the spike variability dominates the hardware variability in most of the cases. Our results can be further improved by automating the tuning process of the circuits responsible for emulating the calcium dynamics. Future steps include emulating full networks that obey the STC plasticity rule which requires mapping the synaptic weights to hardware and further use of stochastic rounding. Our work highlights the power of stochastic rounding in overcoming the challenges of reduced precision. Most importantly, it also demonstrates the power of analog neuromorphic hardware such as BrainScaleS-2 in faithfully emulating biological neural dynamics and speeding-up this emulation for research purposes.

Bibliography

Atoui, A. (2024). Memory consolidation and synaptic plasticity on the neuromorphic brainscales-2 system. Technical report, University of Tübingen.

Billaudelle, S. (2022). *From transistors to learning systems: Circuits and algorithms for brain-inspired computing.* Phd dissertation, University of Heidelberg.

Billaudelle, S., Cramer, B., Petrovici, M. A., Schreiber, K., Kappel, D., Schemmel, J., and Meier, K. (2021). Structural plasticity on an accelerated analog neuromorphic hardware system. *Neural Networks*, 133:11–20.

Billaudelle, S., Weis, J., Dauer, P., and Schemmel, J. (2022). An accurate and flexible analog emulation of adex neuron dynamics in silicon. *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 23:1–4.

Bohnstingl, T., Scherr, F., Pehle, C., Meier, K., and Maass, W. (2019). Neuromorphic hardware learns to learn. *Frontiers in Neuroscience*, 13.

Brette, R. and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J Neurophysiol*, 94(5):3637–3642.

Chen, L. (1995). Testing the mean of skewed distributions. *Journal of the American Statistical Association*, 90(430):767–772.

Croci, M., Fasi, M., Higham, N. J., Mary, T., and Mikaitis, M. (2022). Stochastic rounding: Implementation, error analysis and applications. *Royal Society Open Science*, 9(3).

Douglas, R., Mahowald, M., and Mead, C. (1995). Neuromorphic analogue vlsi. *Annual Review of Neuroscience*, 18:255–281. Funding by Office of Naval Research (ONR).

Forsythe, G. (1950). Round-off errors in numerical integration on automatic machinery. *Bulletin of the American Mathematical Society.*

Frey, U. and Morris, R. G. (1997). Synaptic tagging and long-term potentiation. *Nature*, 385(6616):533–536.

Friedmann, S., Schemmel, J., Gruebl, A., Hartel, A., Hock, M., and Meier, K. (2016). Demonstrating hybrid learning in a flexible neuromorphic hardware system.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York.

Hopkins, M., Mikaitis, M., Lester, D. R., and Furber, S. (2020). Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ordinary differential equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2166):20190052.

Indiveri, G. and Liu, S. (2015). Memory and information processing in neuromorphic systems. *CoRR*, abs/1506.03264.

Kong, Q., Siauw, T., and Bayen, A. M. (2021). *Python programming and numerical methods: A guide for engineers and scientists.* Academic Press, an imprint of Elsevier.

Lamprecht, R. and LeDoux, J. (2004). Structural plasticity and memory. *Nature Reviews Neuroscience*, 5(1):45–54.

Luboeinski, J. and Tetzlaff, C. (2021). Memory consolidation and improvement by synaptic tagging and capture in recurrent neural networks. *Communications Biology*, 4(1):275.

Marsaglia, G. (2003). Xorshift rngs. *Journal of Statistical Software*, 8(14):1–6.

Okuda, K., Højgaard, K., Privitera, L., Bayraktar, G., and Takeuchi, T. (2021). Initial memory consolidation and the synaptic tagging and capture hypothesis. *European Journal of Neuroscience*, 54(8):6826–6849.

Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., Weis, J., Leibfried, A., Müller, E., and Schemmel, J. (2022). The brainscales-2 accelerated neuromorphic system with hybrid plasticity.

Redondo, R. L. and Morris, R. G. M. (2011). Making memories last: the synaptic tagging and capture hypothesis. *Nature Reviews Neuroscience*, 12(1):17–30.

Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., and Kay, B. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19.

Tetzlaff, C., Kolodziejski, C., Markelic, I., and Wörgötter, F. (2012). Time scales of memory, learning, and plasticity. *Biol Cybern*, 106(11-12):715–726.

Wang, S.-H., Redondo, R. L., and Morris, R. G. (2010). Relevance of synaptic tagging and capture to the persistence of long-term potentiation and everyday spatial memory. *Proceedings of the National Academy of Sciences*, 107(45):19537–19542.

Weis, J. (2020). Inference with artificial neural networks on neuromorphic hardware. Master's thesis, Heidelberg University.

Wunderlich, T., Kungl, A. F., Müller, E., Hartel, A., Stradmann, Y., Aamir, S. A., Grübl, A., Heimbrecht, A., Schreiber, K., Stöckel, D., Pehle, C., Billaudelle, S., Kiene, G., Mauch, C., Schemmel, J., Meier, K., and Petrovici, M. A. (2019). Demonstrating advantages of neuromorphic computation: A pilot study. *Frontiers in Neuroscience*, 13.

Zenke, F. and Gerstner, W. (2014). Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Frontiers in Neuroinformatics*, 8.

Appendix

A.1 Calcium Mapping

We present the calcium mapping results for the chips that were used for comparison of emulation results in figures A.1.1 and A.1.2.

A.2 Single Trial Behavior

The trial-to-trial variability can differ depending on the chips and spikes as shown in figure A.2.1.

A.3 Different Spikes

Figures A.3.1, A.3.2, and A.3.3 show the emulation results for different chips where the emulation is compared against a simulation carried out at a time step of 0.2 ms.

A.4 Information on Data Storage and Software

Here we present information on the software that was generated for this thesis for the implementation of the plasticity rules and running experiments. The experiments were executed in the container environment latest == 2024-04-17_1.img.

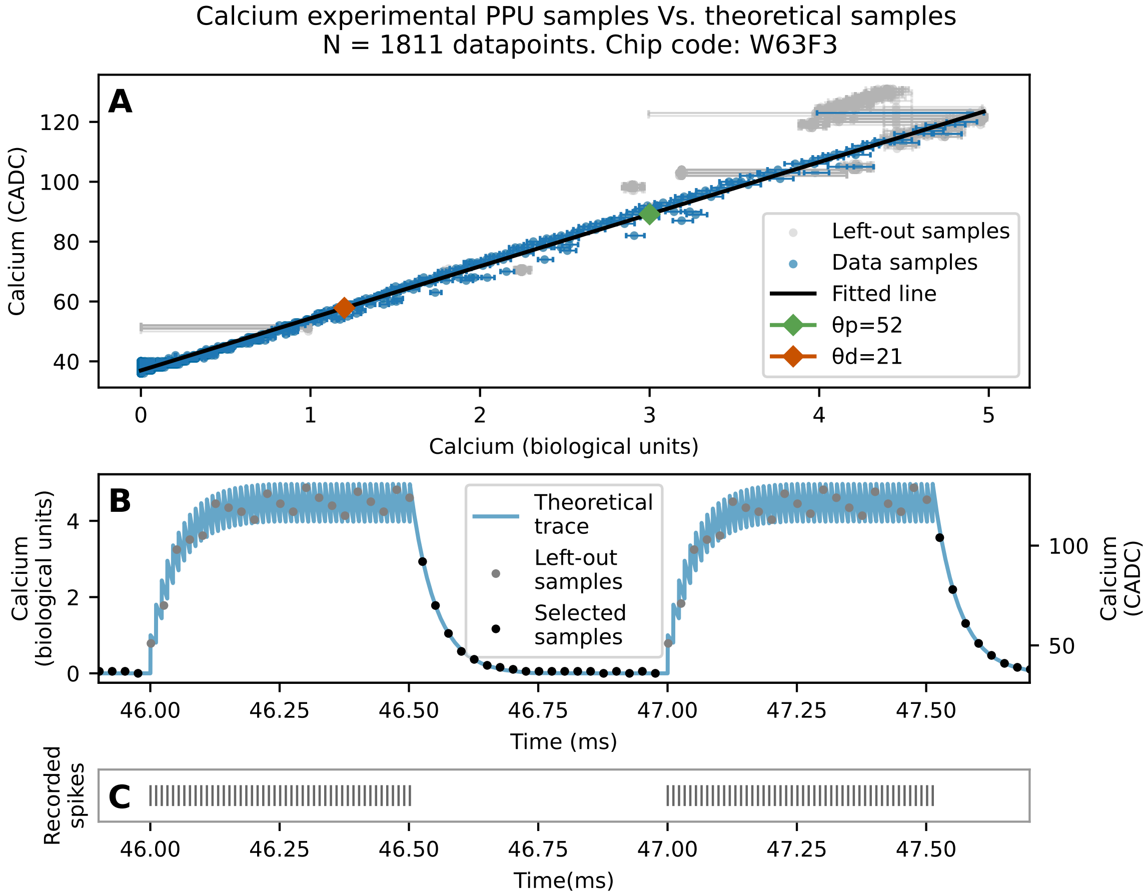


Figure A.1.1: Mapping the calcium potentiation and depression thresholds using linear regression and least-squares estimation. (A) Calcium column-parallel analog-to-digital converter (CADC) values of the samples collected from running the calcium mapping experiment versus theoretical calcium calculated using eq. (3.4) at the same time instants of the samples. The error bars are the expected errors from the variability in time samples. Samples with low expected errors are selected and fitted using least-squares method to estimate the potentiation and depression thresholds. (B) Theoretical calcium trace calculated from the recorded spikes of the parrot neuron using eq. (3.4) and plasticity processing unit (PPU) samples. The left-out samples are in the spiking duration of the mapping protocol whereas the selected samples are in the decaying region of the mapping protocol. (C) Recorded spikes of the parrot neuron used to generate the theoretical calcium trace and experimental calcium samples.

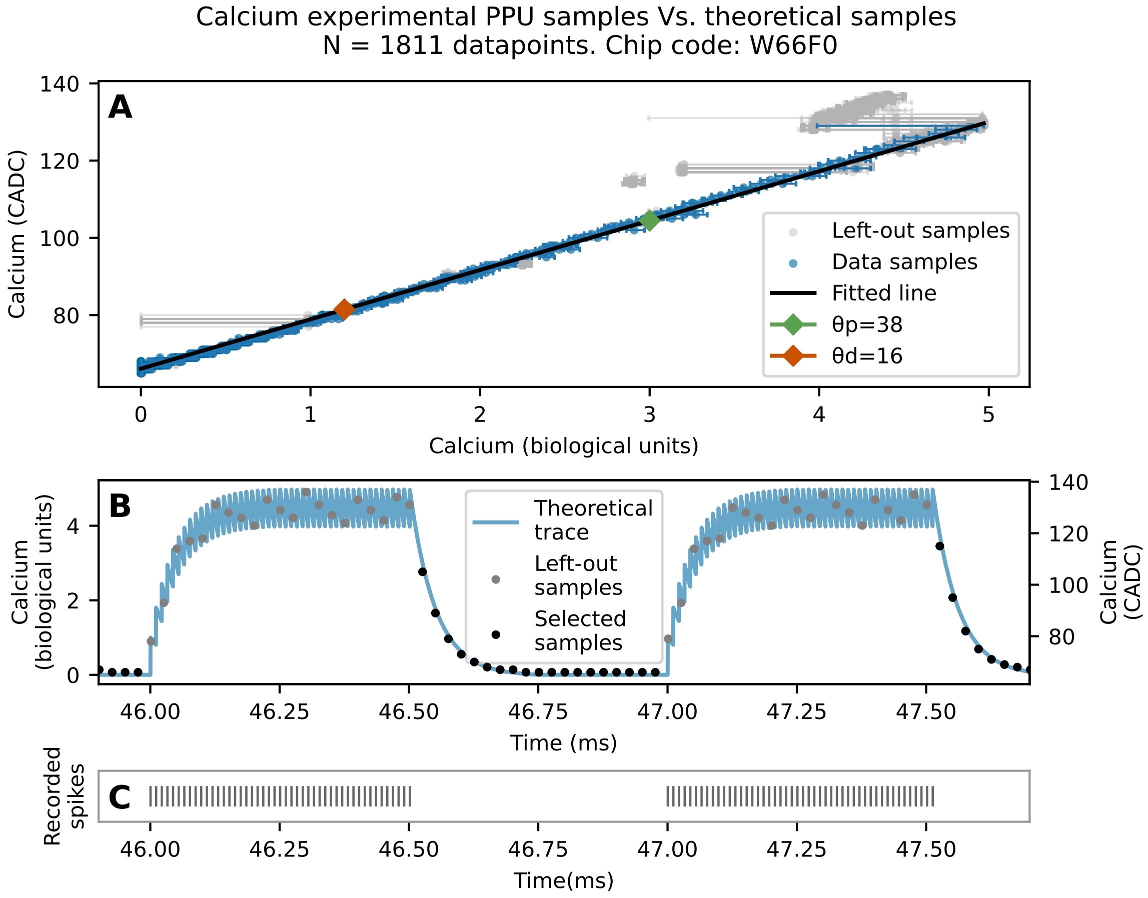


Figure A.1.2: Mapping the calcium potentiation and depression thresholds using linear regression and least-squares estimation. (A) Calcium CADC values of the samples collected from running the calcium mapping experiment versus theoretical calcium calculated using eq. (3.4) at the same time instants of the samples. The error bars are the expected errors from the variability in time samples. Samples with low expected errors are selected and fitted using least-squares method to estimate the potentiation and depression thresholds. (B) Theoretical calcium trace calculated from the recorded spikes of the parrot neuron using eq. (3.4) and PPU samples. The left-out samples are in the spiking duration of the mapping protocol whereas the selected samples are in the decaying region of the mapping protocol. (C) Recorded spikes of the parrot neuron used to generate the theoretical calcium trace and experimental calcium samples.

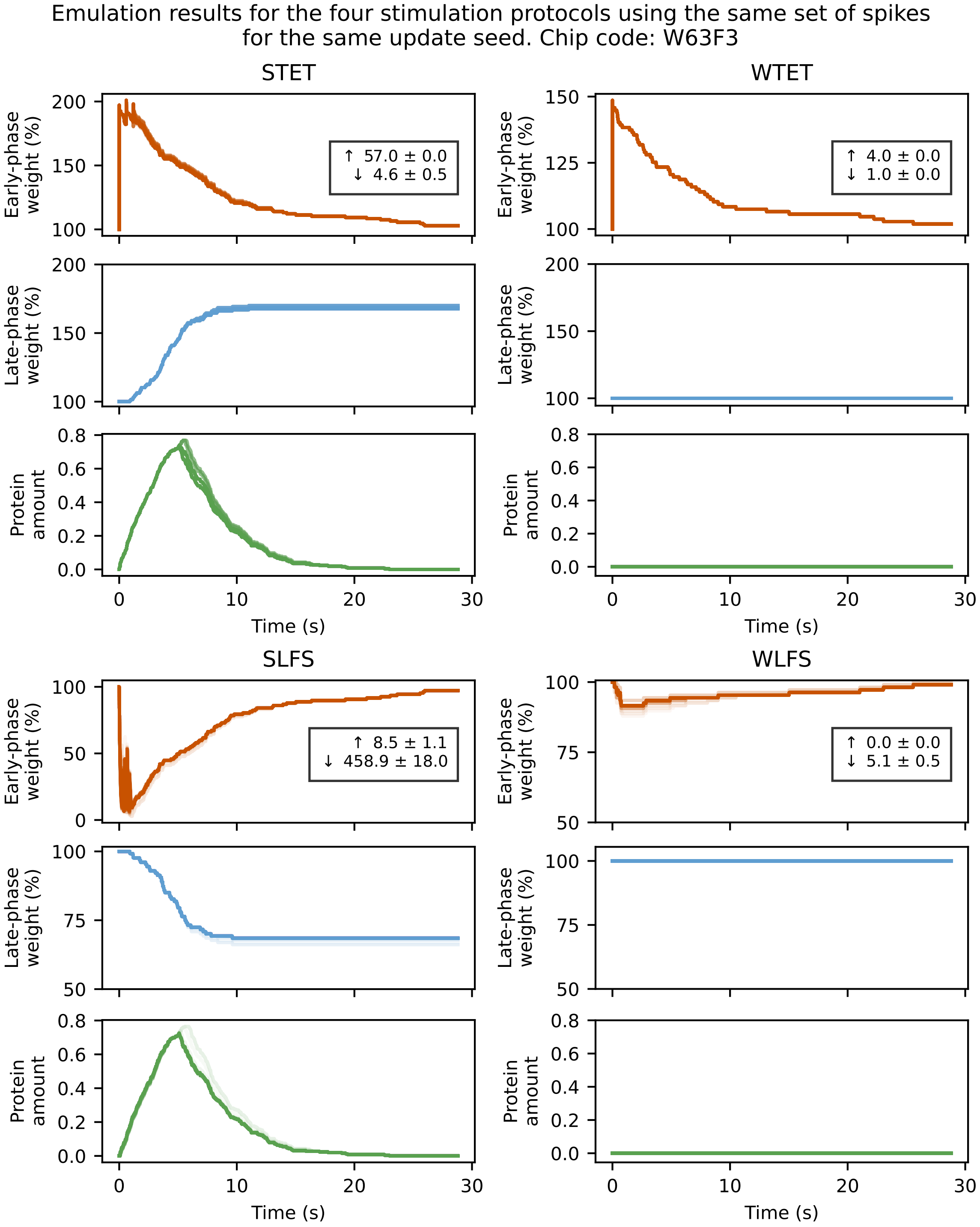


Figure A.2.1: Emulation results for the four stimulation protocols for one set of spikes repeated 100 times for a single update seed. The aim is to check the reproducibility of the results over different trials. The source of variability is the adaptation circuit. The measures of performance are the mean and standard deviation of the number of potentiations and depressions shown by arrows pointing upwards and downwards respectively for each protocol.

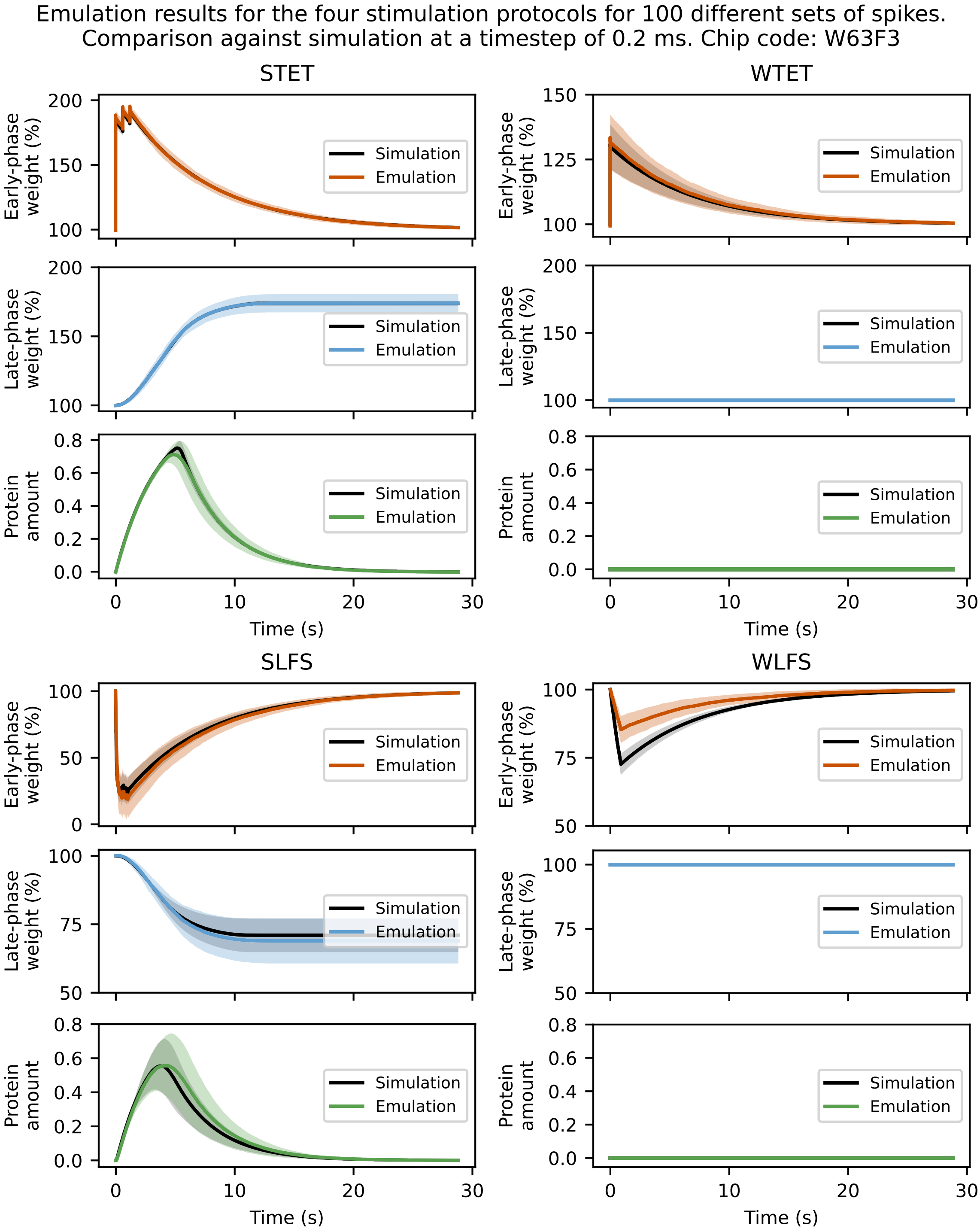


Figure A.3.1: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 0.2 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

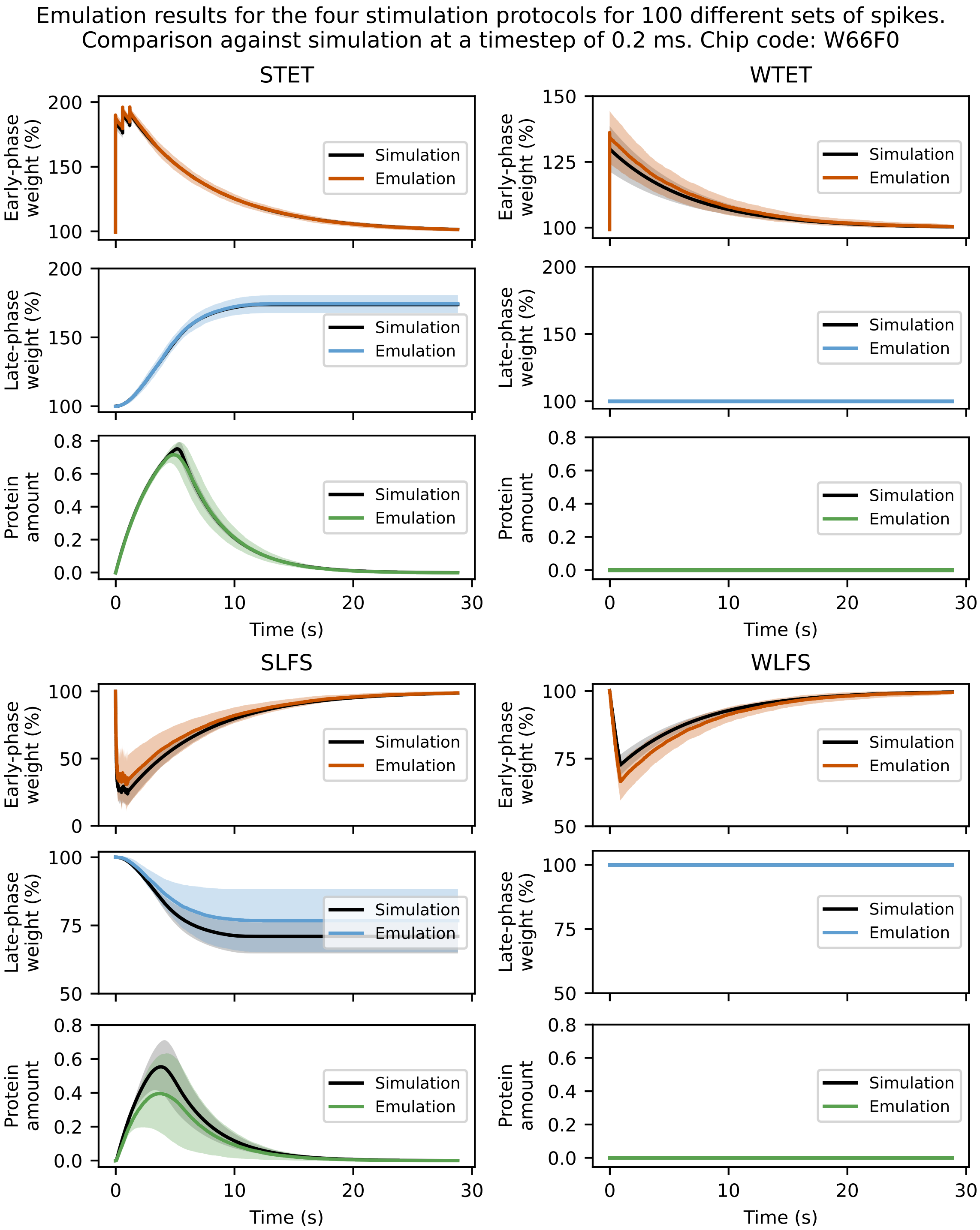
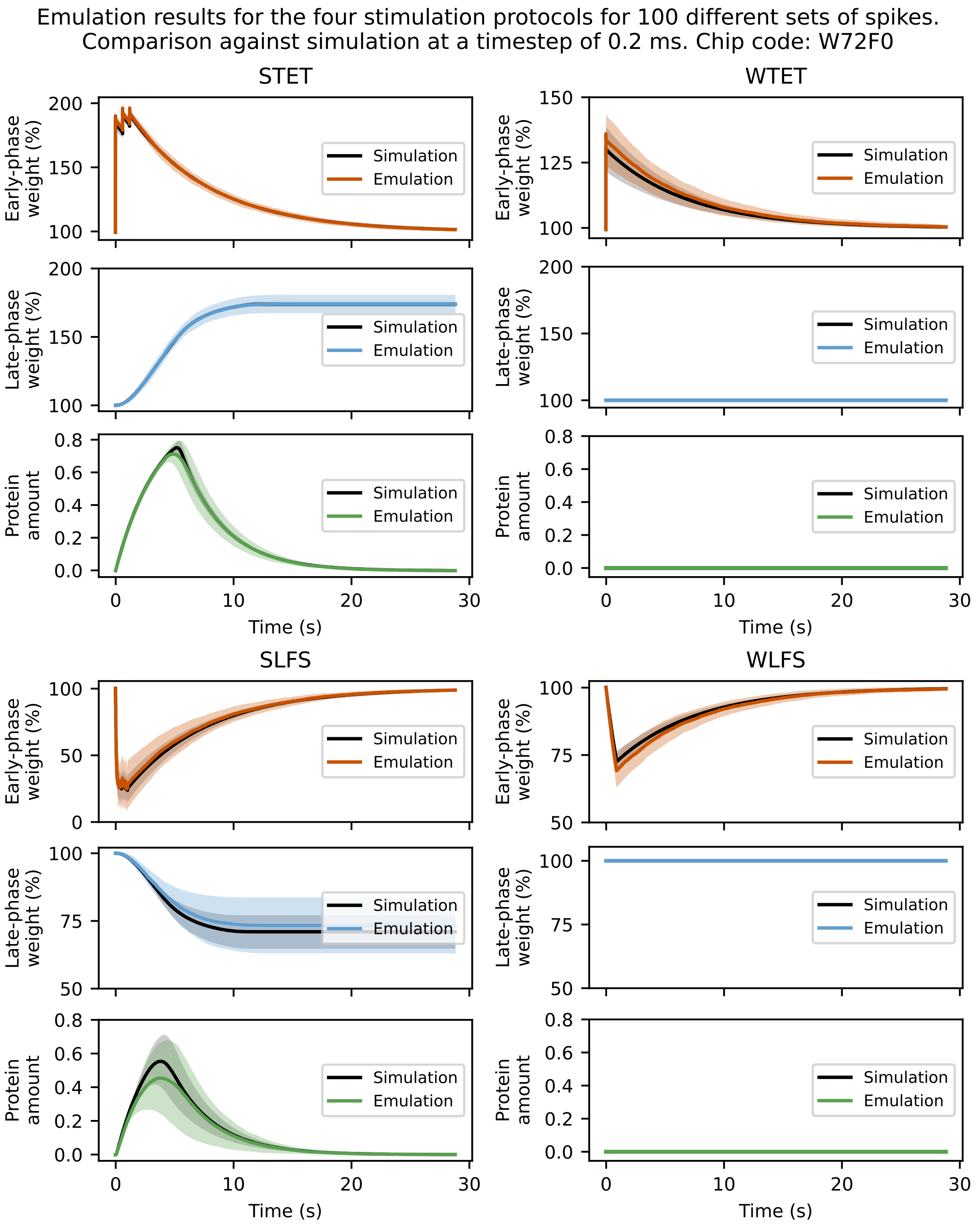


Figure A.3.2: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 0.2 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.



[!ht]

Figure A.3.3: Emulation results for the four stimulation protocols compared against a simulation baseline at a time step of 0.2 ms for 100 different sets of spikes. The lines correspond to the average early-phase and late-phase weights and protein amount. The bands correspond to one standard deviation from the average. For comparison purposes, the simulation results are plotted at an accelerated factor of 1000 to match the time of the emulation results.

Repository	Commit ID
model-hw-memory-consolidation	96a5d6d11e4ba3749652d03da209943049c61ab2
pynn-brainscales	57370c7479f4a4c372ad944c22bec0b5ee30f61b
code-format	09f3a985a6f264359b10a6a129dd6dce7e55c9e8
haldls	237983b173c164d225a2f5398d7e72ef60de7397
grenade	b388b3deb51f09b9871f53e7b5747993d5800e40
calix	a706868c6ba285b1f8fd7cdef1a19d7328e02912
logger	73dadb3ce413c521845ef7d36f818073eee4fefa
halco	a97040a732ab1ba954e077616303a18acf623092
hate	35b3cb211cabbbc5c01036ae7878a73e338166c4
fisch	6120fc0ac0d90b3c66a212b3cc5cc25034bf584e
libnux	66b9c67bc114f82add677c6095f38843c23c4cd7
hxcomm	95abf25670bd8cb7cc5b499cde56f653130cf20c
rant	722edd57c9e42462a660db8a1febb0211ffad07c
ztl	b6745261d8bfdce44516d58d632c3c73834839d2
pywrap	5e2af30e9593882b471d3cd02df00b93f13ff479
lib-boost-patches	136c5b41cb046afe2c726aa4646928bf5190622e
sctrltp	1d854f953f7e8c8ead44406a22bb80421ca3857c
hwdb	c5f86e16b1bb12e2b56d16477867b8369c1fc715
visions-slurm	8f41ea4f5bd1573d8f4623e9ed698a29f30036a3
flange	28e729d59df3b4ff380f84351c40d4da3086bed8
lib-rcf	000185eb11db4d54cb6b12b09af54cf742741036
bss-hw-params	b7be7827b51536804f0bda76f8ba4be693df23a8

Table A.1: Commit IDs of the custom software

Description	Change Sets
Plotting and Evaluation of STC Implementation	23563
PPU implementation using STC plasticity rule	22836
Mapping of calcium thresholds	22960
Minimal PPU example for calcium with plotting	22835
Initializing Plastic Synapse	22834
Minimal MADC example for calcium and plotting	22833

Table A.2: Overview of change sets

Acknowledgements

I happily acknowledge that this work is the fruit of the collective efforts of the Electronic Vision Group at the Kirchhoff Institute for Physics at Heidelberg University.

I would like to thank Dr. Johannes Schemmel for welcoming me into this group, giving me the opportunity to do my lab rotation and thesis, and supervising the project for the past months. Thank you for your trust, constructive feedback, and the opportunity to participate in different events and present my work at different occasions.

I would also like to thank Dr. Sebastian Billaudelle, for his early supervision and building the foundations of this project. Thank you for your input on this project, your fruitful ideas, showing a leading example of an organized and thoughtful researcher, and your support and kindness throughout my journey.

I am extremely thankful to Jakob Kaiser, who supervised me closely since the very beginning and gave detailed feedback regularly. Thank you for always pushing me to improve my work, teaching me new skills, and familiarizing me with new topics. I am grateful for you being involved in this journey and for your patience, kindness, and support, academically and personally.

I want to extend my thanks to Philipp Spilger, for always answering my questions and being involved in overcoming problems related to software implementation and tailoring the needs of this project. I can safely say that the results were made possible from your very helpful input and kind attitude.

I would like to thank every member of this group whether in supervising, supporting the project with their expertise, providing feedback, or even having friendly conversations. Thank you all for setting the high expectations of what a healthy, supporting, and growth-promoting lab environment looks like.

Finally, I am thankful to my family and friends for their endless love and support. I would like to thank my mother for empowering me as always and my father for pushing me beyond my comfort zone and encouraging me to never stop learning. I am very grateful for my brothers, for always having my back, especially my elder brother who supported me personally and financially throughout my stay abroad. I am also grateful for the support of my friends, especially Miriam, my favourite person in Tübingen.

The work carried out in this Master's thesis used systems, which received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreements Nos. 720270, 785907 and 945539 (Human Brain Project, HBP) and Horizon Europe grant agreement No. 101147319 (EBRAINS 2.0).