

**Fakultät für Physik und Astronomie**  
**Ruprecht-Karls-Universität Heidelberg**

**Bachelorarbeit**

im Studiengang Physik

vorgelegt von

**Arno Friedrich**

geboren in Dresden, Deutschland

**November 2014**



# Charakterisierung von Adaption auf neuromorpher Hardware

Diese Bachelorarbeit wurde von

**Arno Friedrich**

ausgeführt am

KIRCHHOFF-INSTITUT FÜR PHYSIK

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

unter der Betreuung von

**Prof. Dr. Karlheinz Meier**



## ZUSAMMENFASSUNG

Im Rahmen dieser Arbeit wird die Implementierung des Adaptionverhaltens eines neuromorphen Hardwaresystems untersucht. Hierzu werden existierende Messroutinen ob ihrer Nutzbarkeit überprüft und neue Messverfahren zur Bestimmung von Adaptionparametern präsentiert.

Membranspannungsverläufe des *High-Input-Count-Analog-Neural -Network-Chips* (HICANN) werden dabei untersucht und die gewonnenen Ergebnisse mit theoretischen Vorgaben verglichen. Gleichzeitig wird eine mathematische Analyse des dargestellten Neuronen-Modells durchgeführt, womit ein tieferer Einblick in die zugrunde liegenden Funktionsweisen von Adaption ermöglicht wird.

Des Weiteren wurde ein Simulationsprotokoll entwickelt, mit dem Aussagen über die Genauigkeit und Grenzen der implementierten Messalgorithmen getroffen werden konnten. Somit war es erstmals möglich, das Verhalten der Adaptionzeitkonstante sowie des Adaptionfaktors in Abhängigkeit von den zugehörigen Biasströmen zu ermitteln und die gewonnenen Ergebnisse den Designvorgaben gegenüberzustellen.

## ABSTRACT

In this thesis, the implementation of adaptation on a neuromorphic hardware system is characterised. For that purpose, the feasibility of existing measurement routines is evaluated and new approaches on the determination of adaptation parameters are presented. Membrane voltage curves from the HICANN-Chip are examined and the obtained results are compared with theoretical models. Furthermore a mathematical analysis of the implemented neuron model was conducted, to gain insight into the underlying principles of adaptation. In order to validate the measurement techniques, a simulation protocol was developed, which resulted in statements about the accuracy and limitations of the proposed techniques.

As a conclusion, the observed behaviour of the adaptation time constant and the adaption factor was examined and compared with design specifications of the hardware system.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Grundlagen</b>	<b>2</b>
2.1. AdEx-Modell . . . . .	2
2.2. Adaptionverhalten . . . . .	2
2.3. Adaptionimplementierung . . . . .	3
<b>3. Methoden</b>	<b>4</b>
3.1. Vorgegangene Messverfahren . . . . .	4
3.2. Nutzung von Spikefrequenz Adaption . . . . .	6
3.2.1. Messprinzip . . . . .	6
3.2.2. Überprüfung der Annahmen . . . . .	9
3.3. Regression im <i>subthreshold</i> -Regime . . . . .	13
3.3.1. Berechnung und Testen der Messfunktion . . . . .	13
3.3.2. Messprinzip . . . . .	18
3.3.3. Validierung . . . . .	20
<b>4. Resultate</b>	<b>24</b>
4.1. Messung von $\tau_m$ . . . . .	24
4.2. Messung von $a$ . . . . .	25
4.3. Messung von $\tau_w$ . . . . .	29
<b>5. Diskussion und Ausblick</b>	<b>33</b>
<b>A. Vergleich des Adaptionsfaktors mit Simulationsdaten</b>	<b>36</b>
<b>B. Verwendete Parametersätze</b>	<b>37</b>



# 1. Einleitung

Das menschliche Gehirn ist in der Lage komplexe Zusammenhänge Zeit- und Energieeffizient zu erfassen. Dieses Verhalten und die grundlegenden Prinzipien der Informationsverarbeitung eines neuronalen Netzwerks zu verstehen, ist das Ziel in vielen Forschungsbereichen der Biologie, Informatik und nicht zuletzt Physik.

Ein möglicher Ansatz zur genaueren Analyse dieser Prinzipien ist es, mithilfe von klassischen Recheneinheiten die zugrunde liegenden Differentialgleichungen verschiedener Neuronenmodelle numerisch zu lösen und so deren Dynamik und Implikationen zu betrachten (*EPFL und IBM, 2008*). Ein anderes Vorgehen wird innerhalb des BrainScaleS Projekt (*BrainScaleS, 2012*) untersucht. Hierbei werden analoge Schaltkreise entwickelt, welche die einzelnen Komponenten der physikalischen Modelle wiedergeben können. Diese werden in Verbindung mit einer digitalen Event-Auslese auf so genannten *High-Input-Count-Analog-Neural-Network-Chips* (HICANN-Chips) (*HBP SP9 partners, 2014*) implementiert. Der Vorteil eines solchen Ansatzes ist ein geringer Energieverbrauch pro Event und eine wesentlich erhöhte Ausführungsgeschwindigkeit im Vergleich zu herkömmlichen Simulationsmodellen. Außerdem ist die Skalierung der darstellbaren neuronalen Netzwerke nicht von vorhandener Rechenleistung begrenzt. Um großskalige Simulationen zu ermöglichen wurde dabei auf Wafer Scale Integration gesetzt (*Schemmel et al., 2008*).

Für ein stabiles Verhalten der untersuchten Netzwerke ist es entscheidend, dass die einzelnen Komponenten der Hardwareimplementierung über dem gesamten HICANN-Chip ein ähnliches Verhalten zeigen. Aufgrund von Variation im Fertigungsprozess, müssen daher Verfahren entwickelt werden, die Schwankungen der Hardwareparameter erfassen und gegebenenfalls korrigieren können. In *Binh (2013)* wurde hierzu eine Demonstrationssoftware entwickelt, die es erlaubt definierte Eventmuster auf einem einzelnen Neuron wiederzugeben. Die beschriebenen Variationen wurden dabei durch manuelle Tests ausgeglichen. Das Ziel muss es jedoch sein, diesen Vorgang zu automatisieren um eine HICANN-Chip weite Realisierung der erwünschten Eventmuster zu ermöglichen. Ein Beispiel für ein solches Muster ist die Adaption des Feuerrate bei dauerhaft wiederkehrenden Aktionspotentialen.

In der vorliegenden Arbeit wurden erste Messmethoden entwickelt, um eine verlässliche Konfiguration dieses Effektes auf dem gesamten HICANN-Chip zu ermöglichen. Hierbei wurden bereits beschriebene Verfahren auf ihre Nutzbarkeit für Messungen auf dem HICANN-Chip analysiert und ein neues Messprinzip zur Bestimmung der zwei Parameter: Adaptionsfaktor und Adaptionszeitkonstante entwickelt. Gleichzeitig wurde ein Simulationsprotokoll zur Überprüfung der Genauigkeit der entworfenen Messmethode implementiert. Zum Abschluss sind erste Messungen dieser Hardwareparameter durchgeführt, und die gewonnen Erkenntnisse mit den theoretischen Erwartungen verglichen worden.

## 2. Grundlagen

Im Gegensatz zu Simulationsumgebungen auf herkömmlichen Recheneinheiten muss bei der Entwicklung von neuromorpher Hardware das darzustellende Neuronenmodell von vornherein festgelegt werden. Im folgenden wird das hier genutzte Modell und seine Implementierung auf der Hardware beschrieben. Dabei werden die Funktionen und Eigenschaften des adaptiven Terms mit besondere Aufmerksamkeit betrachtet.

### 2.1. AdEx-Modell

Auf dem HICANN-Chip ist das *Adaptive-Exponential-Integrate-and-Fire-Modell* (AdEx) (Brette und Gerstner, 2005) implementiert. Dieses beschreibt den Verlauf der Membranspannung eines Neurons mithilfe von zwei gekoppelten Differentialgleichungen:

$$C_m \frac{dV}{dt} = -g_l(V - E_l) + g_l \cdot \Delta_T \cdot \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I(t) \quad (1)$$

$$\tau_w \cdot \frac{dw}{dt} = a \cdot (V - E_l) - w. \quad (2)$$

Wobei folgende Bedingungen gelten,

$$\text{wenn: } V(t) = V_{\text{th}} \quad \Rightarrow \quad \begin{aligned} V(t) &\rightarrow V_{\text{reset}} \\ w(t) &\rightarrow w + b. \end{aligned}$$

Hierzu stellt  $C_m$  die Membrankapazität,  $E_l$  das Leckpotential und  $g_l$  die Leckkonduktanz dar. Zudem beschreibt  $V_{\text{th}}$  das Threshold- und  $V_{\text{reset}}$  das Reset-potential. Durch den Schwellenwert  $V_T$  und den Steigungsfaktor  $\Delta_T$  kann ein starker exponentieller Anstieg, sobald das Membranpotential einen einstellbaren Spannungswert erreicht, emuliert werden.  $I(t)$  kennzeichnet ein zeitabhängiges Eingangssignal. Die Adaptionzeitkonstante  $\tau_w$ , der Adaptionsfaktor  $a$  und der *spike-triggered* Adaptionsfaktor  $b$  charakterisieren gemeinsam den adaptiven Term des Modells.

### 2.2. Adaptionsverhalten

Der Einfluss dieses adaptiven Terms kann prinzipiell in zwei verschiedenen Regimes des Neurons besonders deutlich beobachtet werden. Im Folgenden wird daher eine Neuronenkonfiguration bei der das Membranpotential durchgängig unterhalb von  $V_{\text{th}}$  liegt als *subthreshold-Regime* und wenn das Ruhepotential  $E_l$  sich oberhalb von  $V_T$ , bzw bei ausgeschaltetem exponentiellen Term  $V_{\text{th}}$ , befindet als *continuous-spiking-Regime* bezeichnet. Allgemein lassen sich mithilfe von Adaptionseffekten eine Vielzahl von verschiedenen Feuermustern realisieren. Darunter sind beispielsweise: *initial-bursting*, *transient-spiking* oder *spike-frequency-adaptation* (Gerstner und Kistler, 2002). Exemplarisch für die Funktionsweise von Adaption ist in Abbildung 1 eine *NEST-Simulation* (Gewaltig und Diesmann, 2007) des Feuermusters *spike-frequency-adaptation* dargestellt. Dabei wurden zwei externe Stromstimuli mit der Form von Rechteckpulse simuliert. Gezeigt wird hier der Verlauf der Membranspannung sowie des Stroms  $w$ .

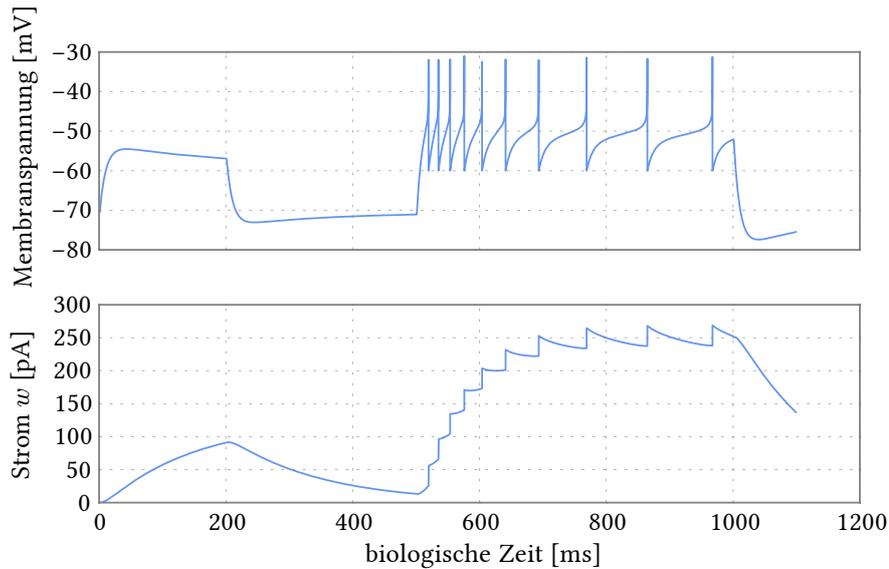


Abbildung 1: *NEST*-Simulation von Membranspannung und Strom  $w$  bei eingeschaltetem adaptiven und exponentiellen Term. Externe Stromstimuli wurden zwischen 0 und 200 ms sowie zwischen 500 und 1000 ms simuliert. Es wird dabei das Feuermuster *spike-frequency-adaptation* dargestellt.

Bei *NEST* handelt es sich um eine Simulationsumgebung für spikende neuronale Netzwerke. Diese löst die zugrunde liegenden Differentialgleichungen des AdEx-Modells numerisch und erlaubt es so, die theoretische Membranspannung eines voll konfigurierbaren Adex-Neurons darzustellen.

Man erkennt nun, dass die Spikelfrequenz im *continuous-spiking*-Regime mit steigendem Strom  $w$  abfällt. Im *subthreshold*-Regime, das zwischen 0 und 500 ms visualisiert ist, beeinflussen die Adaptionparameter  $a$  und  $\tau_w$  den Verlauf der Membranspannung bei Änderung des Ruhepotentials.  $V_{\text{mem}}$  bewegt sich dabei nicht mehr asymptotisch auf das Ruhepotential zu, sondern es findet eine Überschwingung statt. Die Stärke und Periodendauer der gedämpften Schwingung ist abhängig von  $a$  und  $\tau_w$ . Dieser Effekt spielt eine entscheidende Rolle bei der Entwicklung von Messmethoden und wird in Abschnitt 3.3 genauer untersucht.

### 2.3. Adaptionimplementierung

Die Implementierung des Adaptionsverhalten auf dem HICANN-Chip wird durch eine in *Millner (2012)* entwickelten Analschaltung realisiert. In Abbildung 2 ist diese schematisch dargestellt.

Die Adaptionszeitkonstante  $\tau_w$  hängt hierbei indirekt proportional von der Konduktanz vom gegen gekoppelten  $\text{OTA}_w$  ab. Dadurch kann sie über den Biasstrom  $I_{\text{radapt}}$  kontrolliert werden. Der Adaptionsfaktor  $a$  entspricht direkt der Konduktanz von  $\text{OTA}_a$ .

Die Größe von  $b$  wird durch den Hardwareparameter  $I_{\text{fire}}$  eingestellt. Über  $\text{fire}$  wird ein Puls auf das Gate des angeschlossenen Transistor geleitet wenn ein Spike in der Membranspannung registriert wurde. In diesem Fall wird eine über  $I_{\text{fire}}$  kontrollierbare Ladung auf den Kondensator  $C_w$  geleitet.



verbinden.

$$\tau_m \cdot \frac{dV}{dt} = -\left(1 + \frac{a}{g_l}\right) \cdot (V - E_l) \quad (5)$$

Kombiniert mit der Anfangsbedingung  $V(t=0) = V_{\text{reset}}$ , hat dies eine eindeutige Lösung von der Form:

$$V(t) = (V_{\text{reset}} - E_l) \cdot \exp\left(-\frac{1 + \frac{a}{g_l}}{\tau_m} \cdot t\right) + E_l \quad (6)$$

Wenn nun das Neuron in einen durchgehend spikenden Zustand gebracht wird, indem man die Thresholdspannung  $V_{\text{th}}$  über die Leckspannung  $E_l$  setzt, so kann eine weitere Bedingung ausgenutzt werden. In diesem Fall gilt:  $V(t=T) = V_{\text{th}}$ . Wobei  $T$  die Periodendauer der Spikes darstellt. Und es kann so eine Abhängigkeit zwischen der Spikefrequenz des Neurons und dem Adaptionfaktor  $a$  berechnet werden:

$$V_{\text{th}} = (V_{\text{reset}} - E_l) \cdot \exp\left(\frac{1 + \frac{a}{g_l}}{\tau_m} \cdot T\right) + E_l \quad (7)$$

$$\Rightarrow T = \ln\left(\frac{V_{\text{th}} - E_l}{V_{\text{reset}} - E_l}\right) \cdot \frac{\tau_m}{1 + \frac{a}{g_l}} \quad (8)$$

Es wurde in *Schwartz (2013)* nun davon ausgegangen, dass alle relevanten Neuronenparameter außer  $a$  bekannt sind. Daraufhin konnte die Spikefrequenz gemessen und mit Daten von einer separaten Simulationsebene des AdEx-Modells mit gleichen Parametern verglichen werden, um den Wert der Konduktanz  $a$  zu bestimmen.

Dabei wird davon ausgegangen, dass die Implementierung des AdEx-Modells auf Hardware exakt dem Modell in der Simulation entspricht. Wie auf dem Schaltplan in Abbildung 2 zu sehen ist, sind die Differentialgleichungen 7 und 8 jedoch mithilfe von OTAs implementiert. Diese haben einen linearen Bereich von ca. 100 mV (*Millner, 2012, S. 40*). Daher ist die Annahme, dass die Hardware den Vorgaben des Modells folgt, nur für Spannungswerte in einem Intervall von ca. 100 mV um  $E_l$  gültig. Diese Problematik wird auch in den folgenden Messmethoden bestehen bleiben. Jedoch kann versucht werden durch eine geeignete Wahl des Spannungsbereiches ein möglichst lineares Verhalten der Neuronen zu gewährleisten.

Des Weiteren muss der Parameter  $g_l$  bekannt sein. Dieser konnte bisher jedoch noch nicht mit einer ausreichenden Genauigkeit bestimmt werden (*Schmidt, 2014*). Für eine exakte Ermittlung von  $a$  wäre es daher von Vorteil nicht auf eine separate Simulationsebene ausweichen zu müssen.

Die Messungen der verbleibenden Parameter  $\tau_w$  und  $b$  sind in der Methode von *Schwartz (2013)* abhängig von  $a$ . Aus diesem Grund ist eine möglichst genaue Ermittlung des Adaptionfaktors entscheidend und wird im Folgenden mit besonderer Aufmerksamkeit behandelt.

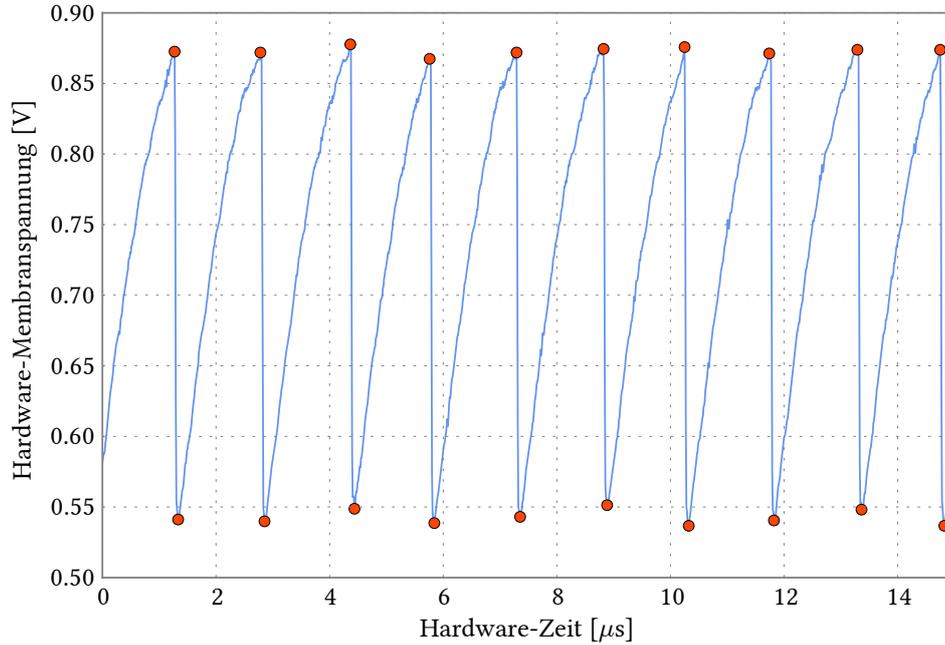


Abbildung 3: Ausschnitt eines Traces zur Bestimmung der Spikeperiode  $T$ ,  $V_{\text{th}}$  und  $V_{\text{reset}}$ . Das Neuron befindet sich in einem durchgehend spikenden Zustand bei ausgeschaltetem exponentiellen Term. Rote Punkte stellen erkannte Spike-Maxima und -Minima des Traces dar. Der eingestellte Parametersatz ist in Tabelle 11 aufgeführt.

## 3.2. Nutzung von Spikefrequenz Adaption

### 3.2.1. Messprinzip

Die Notwendigkeit einer zusätzlichen Simulationsebene kann umgangen werden, indem man die dargestellte Abhängigkeit zwischen  $a$  und  $T$  aus Gleichung 8 direkt nutzt und alle darin enthaltenen zusätzlichen Parameter separat misst. Da die Messung von  $g_l$  auf dem HICANN-Chip nur mithilfe der Beziehung  $\tau_m = \frac{C_m}{g_l}$  möglich ist, ist es sinnvoll  $g_l$  in Gleichung 8 zu ersetzen. Somit ergibt sich für die Beziehung zwischen  $a$  und der Spikeperiode  $T$ :

$$a = \ln \left( \frac{V_{\text{th}} - E_l}{V_{\text{reset}} - E_l} \right) \cdot \frac{C_m}{T} - \frac{C_m}{\tau_m} \quad (9)$$

Es müssen demnach die Parameter  $T$ ,  $V_{\text{th}}$ ,  $V_{\text{reset}}$ ,  $E_l$ ,  $C_m$  und  $\tau_m$  bestimmt werden. Im Folgenden werden diese Parameter exemplarisch durch die Messung von analogen Traces eines Neurons auf dem HICANN-Chip ermittelt.

Die Werte für  $T$ ,  $V_{\text{th}}$  und  $V_{\text{reset}}$  lassen sich innerhalb einer Messung gleichzeitig bestimmen. Hierbei wird die Leckspannung  $E_l$  über die Thresholdspannung  $V_{\text{th}}$  gestellt und somit das Neuron zum kontinuierlichen Spiken gebracht. Nun kann der Spannungsverlauf analog abgegriffen und Maxima sowie Minima des Traces ermittelt werden. In Abbildung

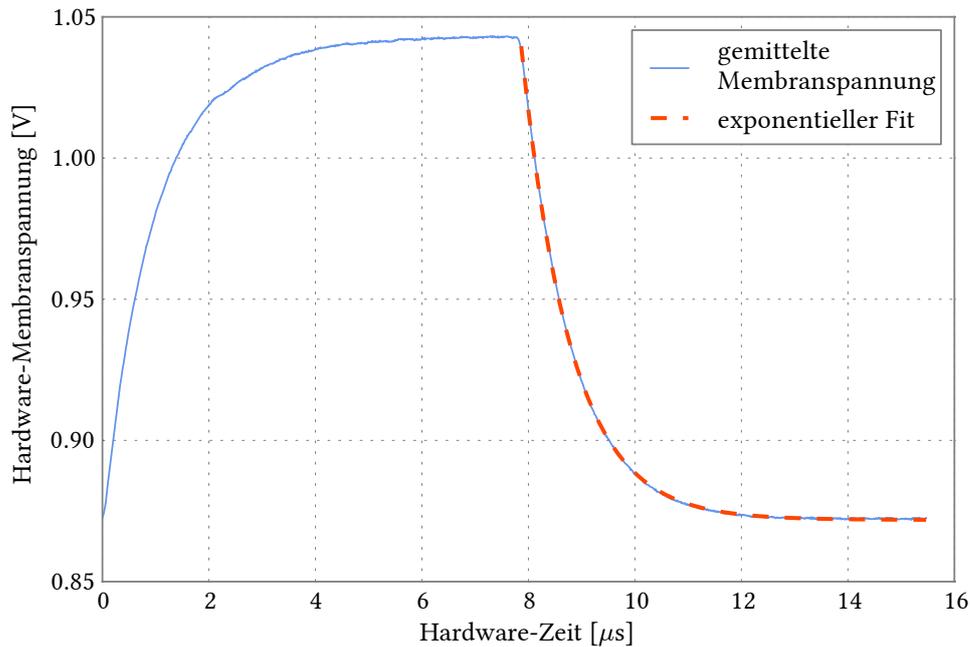


Abbildung 4: Plot der Membranspannung im *subthreshold*-Regime eines Neuron auf dem Vertical Setup *portos*. Durch periodische Rechteckpulse mit einer Frequenz von  $f = 7,8 \mu\text{s}$  und Amplitude von  $I \approx 0,43 \mu\text{A}$  wurden Peaks in der Membranspannung induziert. Diese Spannungsverläufe wurden übereinander gelegt und gemittelt. Der exponentielle Term und Adaption sind so weit wie möglich unterdrückt. Zur Bestimmung der Membranzeitkonstante  $\tau_m$  wurde ein exponentieller Fit an die abfallende Flanke gefittet. Der verwendete Parametersatz kann in Tabelle 9 nachgeschlagen werden.

3 ist ein Ausschnitt des Traces mit markierten Peaks dargestellt. Die Periode  $T$  ist der zeitliche Abstand der so detektierten Spikes. Die Thresholdspannung  $V_{\text{th}}$  wird durch das Maximum eines Spikes dargestellt, wogegen das Minimum die Resetspannung  $V_{\text{reset}}$  wiedergibt. Alle drei Werte wurden über die Spikes des gesamten Traces gemittelt und in Tabelle 1 zusammengefasst. Der für die Messung eingestellte Parametersatz kann in Tabelle 11 nachgeschlagen werden.

Die relative Abweichung der Werte liegt bei maximal  $\Delta V_{\text{reset}}(\text{rel}) = 1,1 \cdot 10^{-3}\%$ . Grund für diesen geringen Fehler ist die hohe Anzahl der aufgenommenen Spikes. Es ist jedoch zu beachten, dass noch keine von *floating-gate-mismatch* verursachte Fehler berücksichtigt werden. Für eine bessere Darstellung der erreichbaren Genauigkeit sollten daher mehrere Messdurchgänge aufgenommen und deren Ergebnisse gemittelt werden.

Die Messung der Leckspannung  $E_l$  erfolgt nach einem ähnlichen, wenn auch einfacheren Prinzip. Hier wird, um jegliches Spikeverhalten zu unterdrücken,  $V_{\text{th}}$  auf die maximale mögliche Spannung angehoben und ein Analogtrace aufgenommen. Die Mittelung der Spannungswerte von diesen Trace ergibt das eingestellte Leckpotential  $E_l$ .

Die Bestimmung der Membranzeitkonstante  $\tau_m$  kann prinzipiell im *subthreshold*-Regime oder im *continuous spiking*-Regime eines Neurons durchgeführt werden (Schwartz, 2013, S.

$T$ [ $\mu\text{s}$ ]	$V_{\text{th}}$ [V]	$V_{\text{reset}}$ [V]
$1,500845 \pm 7 \cdot 10^{-6}$	$0,874570 \pm 6 \cdot 10^{-6}$	$0,544713 \pm 6 \cdot 10^{-6}$

Tabelle 1: Ergebnisse der Bestimmung von  $T$ ,  $V_{\text{th}}$  und  $V_{\text{reset}}$  von einem kontinuierlich spikenden Neuron. Es wurden 667 Spikes in einem Trace mit einer Länge von nominell 1 ms registriert. Die dargestellten Ergebnisse sind die Mittelwerte über allen Spikes. Die Ungenauigkeit stellt den Fehler des Mittelwerts dar. Es muss darauf geachtet werden, dass in den Unsicherheiten noch keine *floatinggate mismatches* berücksichtigt werden.

48). In der vorliegenden Arbeit wird hierfür das *subthreshold*-Regime genutzt. Dazu wird erneut die Thresholdspannung auf den maximalen Wert gesetzt. Das Neuron wird nun mit einem externen Stromstimulus angeregt, der die Form von periodischen Rechteckpulsen hat. Die Reaktion der Membranspannung auf diese Anregung ist gedämpft, weil über die Membrankonduktanz ein Leckstrom abfließt. An die fallenden Flanken, der so entstandenen Peaks in der Membranspannung, kann eine Exponentialfunktion angefitet werden, deren Zeitkonstante  $\tau$  dem gesuchten Parameter  $\tau_m$  entspricht. Die hierfür genutzte Funktion lautet:

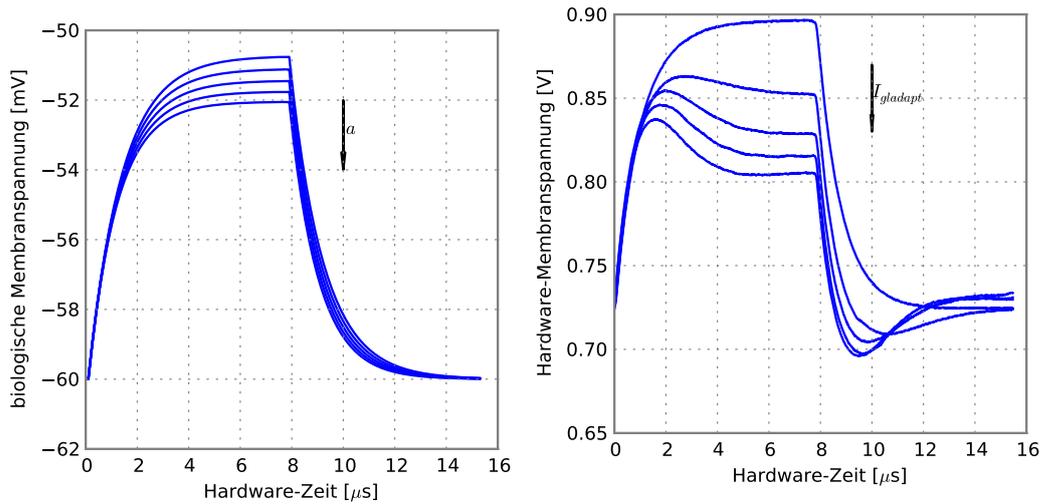
$$V(t) = A \cdot \exp\left(\frac{-t}{\tau}\right) + V_0. \quad (10)$$

Wobei  $A$  der Höhe und  $V_0$  dem Offset der Peaks entspricht.

Bei ungefilterten Analogtraces muss von einem gewissen Rauschen der gemessenen Spannung ausgegangen werden. Um dies als mögliche Fehlerquelle der Fitroutine zu umgehen, wird hier ein Mittelungsverfahren angewendet. Dabei nutzt man den Fakt aus, dass die Periodendauer  $T$  der induzierten Rechteckpulse bekannt ist. Der gemessene Trace kann somit in gleichlange Teile mit der Länge  $T$  geschnitten werden. Diese Tracestücke werden überlagert, wodurch sich das Rauschen weitestgehend heraus mittelt.

Abbildung 4 zeigt einen aus diesem Verfahren entstandenen Peak. An die abfallende Flanke wird Gleichung 10 angefitet und so die Membranzeitkonstante bestimmt. Man erkennt, dass die gewählte Funktion den Fitbereich insgesamt gut darstellen kann. Um die bereits erwähnten *floating-gate-mismatches* zu berücksichtigen, wurden 10 Iterationen der Messung durchgeführt und über die erhaltenen Zeitkonstanten gemittelt. Damit ergibt sich das Ergebnis:  $\tau_m = (0,949 \pm 0.002) \mu\text{s}$ . Die hierfür verwendeten Parameter sind in Tabelle 9 aufgelistet.

Für die Bestimmung der Kapazität  $C_m$  existierte bis zum Zeitpunkt der Niederschrift dieser Arbeit noch keine zuverlässige Methode. In *Schmidt* (2014) wurde jedoch gezeigt, dass der im Design vorgegebene Wert von  $C_m = 2,16 \text{ pF}$  unter der tatsächlichen Membrankapazität liegt. Besonders das Zuschalten der Leitung des externen Stromstimulus bringt eine große parasitäre Kapazität mit sich (*Schmidt*, 2014). Aus diesem Grund kann für  $C_m$  nur eine Abschätzung aus *Millner* (2012, S. 93) verwendet werden, nach der gilt:  $C_m \approx 3,8 \text{ pF}$ . Diese Fehlerquelle wird auch in folgenden Messmethoden der Adaptionsparameter auftreten und muss daher bei der Diskussion von Ergebnissen im Hinterkopf behalten werden.



(a) NEST-Simulation von Membranspannungsverläufen mit  $\tau_w = 0$

(b) gemittelte Messungen auf dem HICANN-Chip mit minimal möglichem  $\tau_w$ , das heißt  $I_{radapt} = 1023$  DAC

Abbildung 5: Vergleich mehrerer Membranspannungen über der Zeit. Links sind NEST-Simulationen und rechts gemittelte Hardware-Traces zu sehen. Die Kurvenscharen beschreiben jeweils die Abhängigkeit der Membranspannung von der sich vergrößernden Konduktanz  $a$  (5a) und dem steigenden Hardware-Parameter  $I_{gladapt}$  (5b). Man erkennt einen deutlichen Einfluss von  $\tau_w$  in der Hardwaremessung in Form eines Overshoots bei größerem  $I_{gladapt}$ . Die biologische Zeit und Zeitkonstanten der Simulation wurden zur besseren Vergleichbarkeit in Hardware-Zeit umgerechnet. Für eine quantitative Analyse müssen die biologischen Spannungen in 5a umskaliert werden. Für eine Darstellung der entscheidenden Effekte ist dies jedoch nicht nötig. Die verwendeten Parametersätze sind in Tabelle 11 aufgelistet.

### 3.2.2. Überprüfung der Annahmen

Der Gedanke liegt nahe, diese Methoden zur Berechnung von  $a$  direkt für Hardwaremessungen zu verwenden. Bevor dies geschehen kann, muss jedoch noch validiert werden ob die Bedingungen des Messprinzips auch von der Hardware erfüllt werden können. Die getroffenen Annahmen lauten wie folgt:

- Annähernd lineares Verhalten der OTAs im Bereich von  $E_l \pm 100$  mV
- vollständige Unterdrückung des exponentiellen Terms
- Einstellung einer gegen 0 gehenden Adaptionzeitkonstante  $\tau_w$

Die entscheidende Bedingung ist hierbei:  $\tau_w \rightarrow 0$ , da die beiden übrigen Punkte bereits in vorherigen Arbeiten gezeigt werden konnten (Denne, 2014).

Um diese Bedingung auf dem HICANN-Chip zu überprüfen, wurden zuerst mehrere Spannungsverläufe nach einem ähnlichen Prinzip wie für die Messung von  $\tau_m$  in Kapitel 3.2.1 aufgenommen. Das heißt, für die Dauer eines Traces wurden periodisch Rechteckpulse von einer externen Quelle auf das Neuron geleitet. Die dadurch erzeugten Peaks in der

Membranspannung mit gleicher Periodizität wurden übereinander gelegt und deren korrespondierende Spannungen gemittelt.

Dabei ist bei allen Messungen die Adaptionzeitkonstante auf den minimal möglichen Wert gestellt. Mit diesem Verfahren werden, für 5 verschiedene Werte des Hardware-Parameter  $I_{gladapt}$ , die Membranspannungen über der Zeit aufgenommen und in Abbildung 5 graphisch dargestellt. Die gleiche Messung wird mithilfe von NEST simuliert und ist in Abbildung 5 auf der linken Seite zu sehen. Auch hier ist  $\tau_w$  auf  $0 \mu s$  gestellt und für 5 verschiedenen Werten des biologischen Parameters  $a$  werden Membranspannungsverläufe simuliert. Da die Zeitkonstanten der Hardware-Neuronen im Vergleich zur Simulation um einen Faktor von  $10^{-4}$  verkürzt sind, wurde zur besseren Anschaulichkeit die biologische Zeit und die biologischen Zeitkonstanten der Simulation in Hardware-Zeit umgerechnet. Mathematisch kann nun der Einfluss von  $\tau_w = 0$  anhand von Gleichung 5 erläutert werden. Hier ist sichtbar, dass aus den grundlegenden Differentialgleichungen des AdEx-Modells bei  $\tau_w = \Delta_T = 0$ , der Parameter  $a$  nur die effektive Konduktanz von  $g_l$  um den konstanten Faktor  $a/g_l$  anhebt. Genau dieses Verhalten lässt sich in Abbildung 5a erkennen. Der Maximalwert der Membranspannung wird bei steigendem  $a$  lediglich herabgesenkt. Die Form des Spannungsverlaufes ändert sich jedoch nicht.

In der Implementierung der Adaption auf dem HICANN-Chip sollte bei  $\tau_w \rightarrow 0$  die Spannung über den in Abbildung 2 mit  $C_w$  gekennzeichneten Kondensator fast verzögerungsfrei der Membranspannung folgen. Dies bedeutet, dass die Adaptionsschaltung im *subthreshold*-Bereich als ein einzelner idealer OTA, der mit der Membranspannung gekoppelt ist, beschrieben werden könnte. Damit würde die Konduktanz  $a$  den gleichen Effekt auf die Membranspannung haben wie die Leckkonduktanz  $g_l$ . Bei einer Adaptionzeitkonstante von  $\tau_w \rightarrow 0$  sollte also die Konduktanz  $a$  den Pullback der Membranspannung durch  $g_l$  lediglich verstärken.

Dieses Verhalten lässt sich durch die in Abbildung 5b dargestellten Messungen nicht bestätigen. Hier wird mit steigendem  $I_{gladapt}$  und dadurch mit steigender Konduktanz  $a$  ein Overshoot der Membranspannung deutlich. Wie in Abschnitt 2.2 beschrieben, ist dies generell ein erwartetes Verhalten für den Adaptionsterm im *subthreshold*-Regime, jedoch wurde in diesem Fall versucht eben jenen Effekt durch ein niedriges  $\tau_w$  zu unterdrücken. Grund für diese Übersteuerung ist ein zeitverzögertes Nachregeln der Spannung  $U_w$  über dem Kondensator  $C_w$  auf das Membranpotential. Die Spannungsdifferenz am OTA  $a$  folgt also nicht mehr instantan der Membranspannung. Dadurch wird auch der das Ausgangssignal von OTA  $a$  zeitlich verzögert. Das heißt, die Hebrabsenkung des Membranpotentials setzt ebenfalls erst verspätet ein. Dieses Verhalten kann in Abbildung 5b beobachtet werden.

Der dynamische Bereich der Adaptionzeitkonstante auf dem HICANN-Chip kann demnach die, für die Messmethode getroffene Annahme, nicht optimal wiedergeben kann. Es bleibt nun zu überprüfen, wie stark diese Abweichung von  $\tau_w$  die Bestimmung der Konduktanz  $a$  beeinflusst und ob der daraus resultierende Fehler klein genug ist um die Methode dennoch anzuwenden.

Zur Abschätzung der möglichen Fehler, die ein  $\tau_w > 0$  verursacht, kann erneut die Simulationssoftware NEST genutzt werden. Dazu wurde die in Abschnitt 3.2.1 beschriebene Messmethode verwirklicht und mit Spannungsverläufen gespeist, die in der Simulation erzeugt wurden. Der Vorteil liegt hier in der Möglichkeit den Wert der Konduktanz  $a$  exakt einstellen zu können, und somit als Referenz für die Präzision der Messung zu ver-

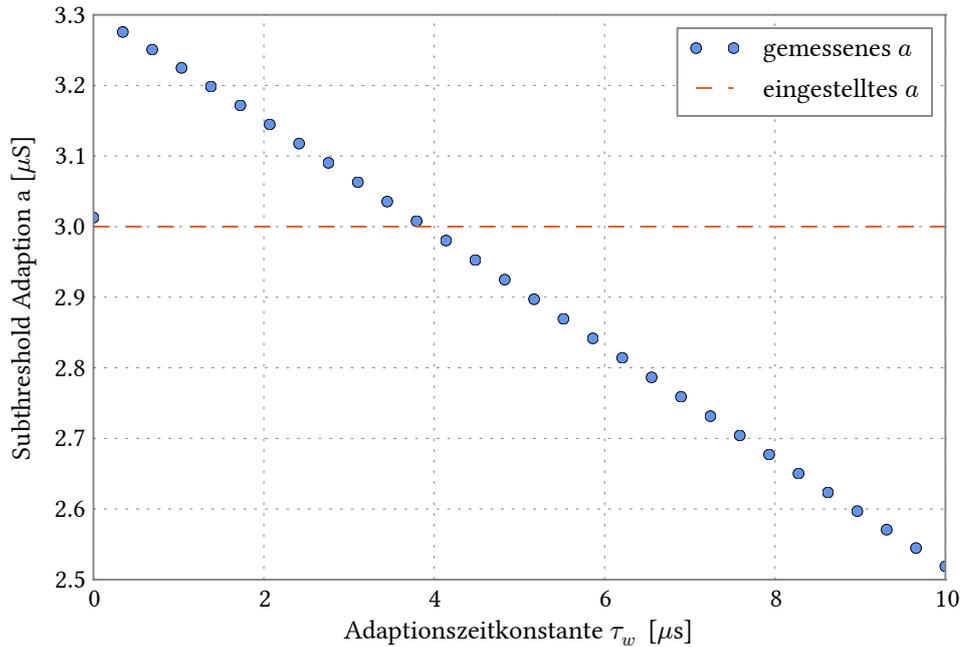


Abbildung 6: Bestimmung der Konduktanz  $a$  durch die in Abschnitt 3.2.1 beschriebenen Methode mithilfe einer *NEST*-Simulation für verschiedene Werte der Adaptionzeitkonstante  $\tau_w$ . Die gestrichelte Linie stellt den in *NEST* eingestellten Wert von  $a$  dar. Es ist eine Abhängigkeit des systematischen Fehlers der Messmethode von  $\tau_w$  erkennbar. Ein vergrößerter Ausschnitt des Intervalls von 0 bis  $0,04 \mu\text{s}$  ist in Abbildung 7 dargestellt. Der für die Messung eingestellte Parametersatz ist in Tabelle 11 zusammengefasst.

wenden.

Dieses Verfahren wurde für mehrere Werte von  $\tau_w$  durchgeführt. In Abbildung 6 ist nun der durch die oben beschriebene Messmethode ermittelte Wert für  $a$  über einen Intervall von eingestellten Werten für  $\tau_w$  dargestellt. Jeder Punkt entspricht dabei einer Messung der Konduktanz  $a$  anhand von simulierten Traces für verschiedene Adaptionzeitkonstanten. Die sonstigen AdEx-Parameter sind für alle Messungen konstant gehalten. Die gestrichelte Linie stellt den in der Simulation eingestellten Wert von  $a$  dar und kann als Referenz für die Genauigkeit der Messung genutzt werden. Man erkennt deutlich, dass eine Abhängigkeit des systematischen Fehlers der Methode von der Adaptionzeitkonstante existiert. Für eine gegen 0 gehende Adaptionzeitkonstante kann der vorgegebene Wert für  $a$  von der Messung mit einem relativen Fehler von 0,4% bestimmt werden. Da es sich hier um eine simulierte Messung handelt, hätte man abgesehen von numerischen Ungenauigkeiten ein exaktes Ergebnis erwartet. Grund für die Abweichung ist, dass in *NEST* keine Zeitkonstanten von genau  $0 \mu\text{s}$  eingestellt werden können. Daher wurde das minimale  $\tau_w$  mit  $0,001 \mu\text{s}$  in Hardware-Zeit festgelegt.

Insgesamt kann man aussagen, dass schon bei einer geringen Erhöhung der Adaptionzeitkonstante sich der systematische Fehler der Messung stark vergrößert. In Abbildung 7 wird dies besonders deutlich. Hier ist ein Ausschnitt der Messung von Abbildung 6 für

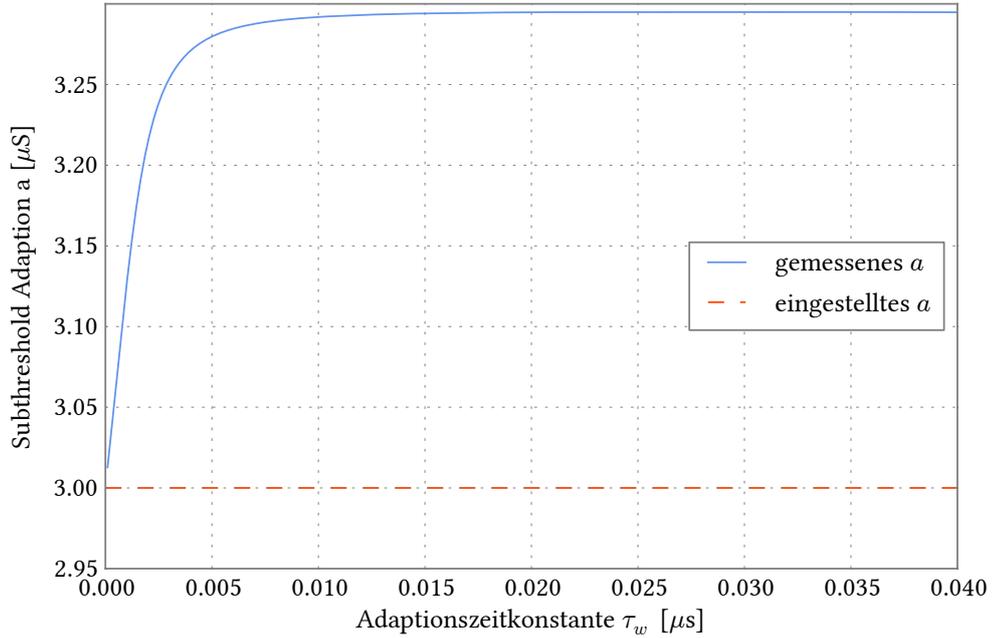


Abbildung 7: Vergrößerung eines Ausschnittes der Messung aus Abbildung 6 für einen Intervall des Parameters  $\tau_w$  von 0 bis 0,04  $\mu\text{s}$ . Die Auflösung der Messpunkte wurde um den Faktor 50 erhöht um den Verlauf der aufgenommenen Daten besser darzustellen.

einen kleinen Bereich von  $\tau_w > 0$  mit einer höheren Auflösung dargestellt. Bei einem an 0 angrenzenden Wert von  $\tau_w \approx 0,005 \mu\text{s}$  erhöht sich die ermittelte Adaption  $a$  bemerkbar. Daraufhin erfolgt ein konstanter Abfall von  $a$ , der in Abbildung 6 zu sehen ist.

Die Abhängigkeit der Messmethode von  $\tau_w$  kann anhand der Grundgleichungen des AdEx-Modells mit unterdrücktem exponentiellen Term erklärt werden. Verbindet man Gleichung 3 und 4 so erhält man:

$$\tau_m \cdot \frac{dV}{dt} = \left(1 + \frac{a}{g_l}\right) \cdot (E_l - V) + \frac{\tau_w}{C_m} \frac{dw}{dt}, \quad (11)$$

mit:  $\tau_m, C_m, a, g_l > 0$ .

Daraus folgt, dass bei  $\tau_w > 0$  die Änderung der Membranspannung durch einen bestimmten Betrag angehoben oder reduziert wird. Außerdem hängt die Periodendauer  $T$  eines Neurons im *continuous spiking*-Regime davon ab wie schnell das Membranpotential die Thresholdspannung erreicht. Es existiert daher folgender Zusammenhang:

$$\frac{1}{T} \sim \frac{dV}{dt}.$$

In der beschriebenen Messmethode wird nun Gleichung 9 genutzt um den Adaptionfaktor  $a$  zu bestimmen. Aus dieser geht hervor:

$$a \sim \frac{1}{T}.$$

Damit ergibt sich die Proportionalität:

$$a \sim \frac{dV}{dt}.$$

In Gleichung 11 wurde jedoch gezeigt, dass für ein  $\tau_w > 0$  die Änderung der Membranspannung von  $\tau_w$  abhängig ist. Wenn man demnach Gleichung 9 zu Bestimmung von  $a$  nutzt und die Adaptionszeitkonstante nicht gegen 0 geht, muss mit einem systematischen Fehler der Messung gerechnet werden. Dieses Verhalten kann in Abbildung 6 beobachtet werden.

Insgesamt kann durch die Simulationen ausgesagt werden, dass der systematische Fehler in der dargestellten Bestimmungsmethode von  $a$  abhängig von der Adaptionszeitkonstante ist, und diese auf dem HICANN-Chip nicht auf einen vernachlässigbar kleinen Wert gestellt werden kann. Außerdem hat die Methode den Nachteil, dass um eine einzelne Messung von  $a$  durchzuführen, der HICANN-Chip für drei verschiedene Messverfahren konfiguriert werden muss. Neben der somit verlängerten Dauer des Experimentes, könnten dadurch *floating-gate*-Variationen einen größeren Einfluss auf die Messgenauigkeit. Allgemein lässt sich also zusammenfassen, dass die beschriebene Methode nicht optimal für Messungen auf dem HICANN-Chip geeignet ist und nach anderen Ansätzen gesucht werden muss.

### 3.3. Regression im *subthreshold*-Regime

Um die im vorherigen Abschnitt angesprochenen Probleme zu umgehen, wird nun eine neue Methode zur Bestimmung von  $a$  und  $\tau_w$  entwickelt.

Der Grundgedanke hier ist es, die Membranspannung eines Neurons im *subthreshold*-Regime analog aufzunehmen und eine von  $\tau_w$  und  $a$  abhängige Funktion  $V(t)$  zu finden, mit der ein solcher Trace angefüttet werden kann. Wenn dies gelingt, hinge die Messung nicht mehr von der Annahme ab, dass die Adaptionszeitkonstante gegen 0 geht. Es wäre zudem möglich  $\tau_w$  und  $a$  mit nur einer Konfiguration der Hardware zu ermitteln.

Der Parameter  $b$  kann hiermit nicht bestimmt werden. Wenn sich die im Folgenden beschriebenen Messalgorithmen jedoch als präzise herausstellen, sollte es möglich das entwickelte Verfahren für die Messung von  $b$  anzupassen. Dabei können Spannungsverläufe im *continous spiking*-Regime eines Neurons genutzt werden, um den Parameter  $b$  in Abhängigkeit von  $a$  und  $\tau_w$  zu bestimmen. Der Aufbau eines solchen Messverfahren wird jedoch kein Teil dieser Arbeit sein.

#### 3.3.1. Berechnung und Testen der Messfunktion

Die Lösung der Differentialgleichungen 3, 4, kann entweder analytisch oder numerisch ermittelt werden. Dabei stellen die Gleichungen 3 und 4 ein inhomogenes gekoppeltes Differentialgleichungssystem erster Ordnung dar. In zusammengefasster Form ergibt sich daraus:

$$\begin{pmatrix} \dot{V}(t) \\ \dot{w}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{-1}{\tau_m} & \frac{-1}{C_m} \\ \frac{a}{\tau_w} & \frac{-1}{\tau_w} \end{pmatrix}}_{\equiv A} \underbrace{\begin{pmatrix} V(t) \\ w(t) \end{pmatrix}}_{\equiv \gamma(t)} + \underbrace{\begin{pmatrix} \frac{E_L}{\tau_m} \\ \frac{-a \cdot E_L}{\tau_w} \end{pmatrix}}_{\equiv b}. \quad (12)$$

Um die allgemeine Lösung für  $\gamma(t)$  zu finden, muss zuerst der homogene Teil von Gleichung 12 bestimmt werden. Das heißt:

$$\dot{\gamma}(t) = A \gamma(t) \quad \text{mit} \quad \gamma(0) = \begin{pmatrix} V_0 \\ w_0 \end{pmatrix}. \quad (13)$$

Wenn  $A$  diagonalisierbar ist, muss nach *Fischer und Kaul* (1990, S. 352) gelten:

$$\gamma(t) = C_1 \cdot e^{\lambda_1 t} \cdot \vec{v}_1 + C_2 \cdot e^{\lambda_2 t} \cdot \vec{v}_2, \quad (14)$$

wobei  $\lambda_{1/2}$  die Eigenwerte und  $\vec{v}_{1/2}$  die Eigenvektoren von  $A$  darstellen. Die Variablen  $C_{1/2}$  beschreiben konstante Vorfaktoren, die abhängig von den Anfangsbedingungen sind.

Es bleibt zu überprüfen, ob die Matrix  $A$  diagonalisierbar ist. Hierfür wurde das Python Package *SymPy* (*Jones et al.*, 2001) verwendet. Tatsächlich konnte die Matrix  $A$  diagonalisiert werden.

Für die Bestimmung der Eigenwerte von  $A$  muss gelten:

$$\det(A - \lambda \mathbb{1}) \stackrel{!}{=} 0. \quad (15)$$

Daraus folgt ein charakteristisches Polynom zweiten Grades, das zu folgenden Eigenwerten führt:

$$\lambda_{1/2} = -\frac{\alpha}{2} \mp \sqrt{\frac{\alpha^2}{4} - \frac{1}{\tau_m \tau_w} - \frac{a}{C_m \tau_w}} \quad \text{mit} \quad \alpha \equiv \frac{\tau_m + \tau_w}{\tau_m \tau_w}. \quad (16)$$

Mithilfe von *SymPy* wurden die dazugehörigen Eigenvektoren bestimmt.

$$\vec{v}_{1/2} = \begin{bmatrix} \frac{2\tau_m \tau_w}{C_m \cdot \left( \tau_m - \tau_w \pm \sqrt{\frac{-4a\tau_w \tau_m^2}{C_m} + (\tau_m - \tau_w)^2} \right)} \\ 1 \end{bmatrix}. \quad (17)$$

Da auf dem HICANN-Chip nur die Membranspannung  $V(t)$  gemessen werden kann, wird im Folgenden die zweite Komponente der Eigenvektoren, und damit die Lösung für  $w(t)$ , nicht weiter betrachtet. Wir können also die Notation dahingehend vereinfachen und schreiben:  $\vec{v}_{1/2} = v_{1/2}^1 = v_i$  mit  $i \in \{1,2\}$ .

Nun folgt die Bestimmung der Koeffizienten  $C_1$  und  $C_2$  aus Gleichung 14. Hierfür muss eine Fallunterscheidung zwischen reellen und komplexen Eigenwerten durchgeführt werden.

#### 1.) $\lambda_i$ reell:

Die Anfangsbedingung aus Gleichung 13 wird in die allgemeine homogene Lösung eingesetzt.

$$\begin{aligned} \Rightarrow \begin{pmatrix} V_0 \\ w_0 \end{pmatrix} &= \begin{pmatrix} C_1 v_1 + C_2 v_2 \\ C_1 + C_2 \end{pmatrix} \quad , \text{weil} \quad v_{1/2}^2 = 1 \\ \Rightarrow C_1 &= \frac{w_0 v_2 - V_0}{v_2 - v_1} \quad , \quad C_2 = \frac{V_0 - w_0 v_1}{v_2 - v_1}. \end{aligned} \quad (18)$$

2.)  $\lambda_i$  komplex:

In diesem Fall muss vor der Bestimmung der Vorfaktoren die homogene Lösung  $\gamma(t)$  noch umgeformt werden. Aufgrund der komplexen Konjugierbarkeit der Eigenwerte und -Vektoren können wir definieren:

$$\lambda_{1/2} \equiv a \mp ib \quad , \quad v_{1/2} \equiv d \pm ie \quad (19)$$

$$\Rightarrow V(t) = C_1(d + ie) \exp(a - ib) + C_2(d - ie) \exp(a + ib) \quad .$$

Wendet man hierauf die Eulersche Formel an und substituiert außerdem  $C_1$  und  $C_2$ , so lautet die Gleichung nun:

$$V(t) = e^{at} \cdot \left[ A (d \sin(bt) + e \cos(bt)) + iB (e \sin(bt) - d \cos(bt)) \right] \quad (20)$$

$$\text{mit } A \equiv C_1 + C_2 \quad , \quad B \equiv C_1 - C_2 \quad .$$

Analog hierzu wird die komplexe Lösung des Stroms  $w(t)$  berechnet.

$$w(t) = e^{at} \cdot [A \cos(bt) - iB \sin(bt)] \quad . \quad (21)$$

Jetzt kann erneut die Anfangsbedingung aus Gleichung 13 genutzt werden, um die Koeffizienten  $A$  und  $B$  zu bestimmen.

$$\Rightarrow A = w_0 \quad , \quad B = \frac{V_0 - w_0 d}{ie} \quad . \quad (22)$$

Nachdem nun alle Komponenten der homogenen Lösung ermittelt wurden, muss zur Lösung von Gleichung 12 noch der partikuläre Teil des Systems berechnet werden. Dazu kann man annehmen das folgende zeitunabhängige Lösung existiert.

$$\gamma_P = \begin{pmatrix} E_0 \\ 0 \end{pmatrix} \quad . \quad (23)$$

Tatsächlich erhält man eine wahre Aussage, wenn wir diese Randbedingung in Gleichung 12 einsetzen und haben damit eine partikuläre Lösung gefunden.

Nun lässt sich die generelle Lösung von  $V(t)$  für reelle und komplexe Eigenwerte zusammenfassen:

$$\text{reell : } V(t) = \frac{(w_0 v_2 - V_0) v_1 e^{\lambda_1 t} + (V_0 - w_0 v_1) v_2 e^{\lambda_2 t}}{v_2 - v_1} + E_l \quad (24)$$

$$\begin{aligned} \text{komplex : } V(t) = e^{at} \cdot & \left[ w_0 (d \sin(bt) + e \cos(bt)) \right. \\ & \left. + \frac{V_0 - w_0 d}{e} (e \sin(bt) - d \cos(bt)) \right] + E_l \quad , \quad (25) \end{aligned}$$

wobei die Variablen  $\lambda_i, v_i, w_0, V_0, a, b, d$  und  $e$  in den Gleichungen 16, 17, 13 und 19 definiert sind.

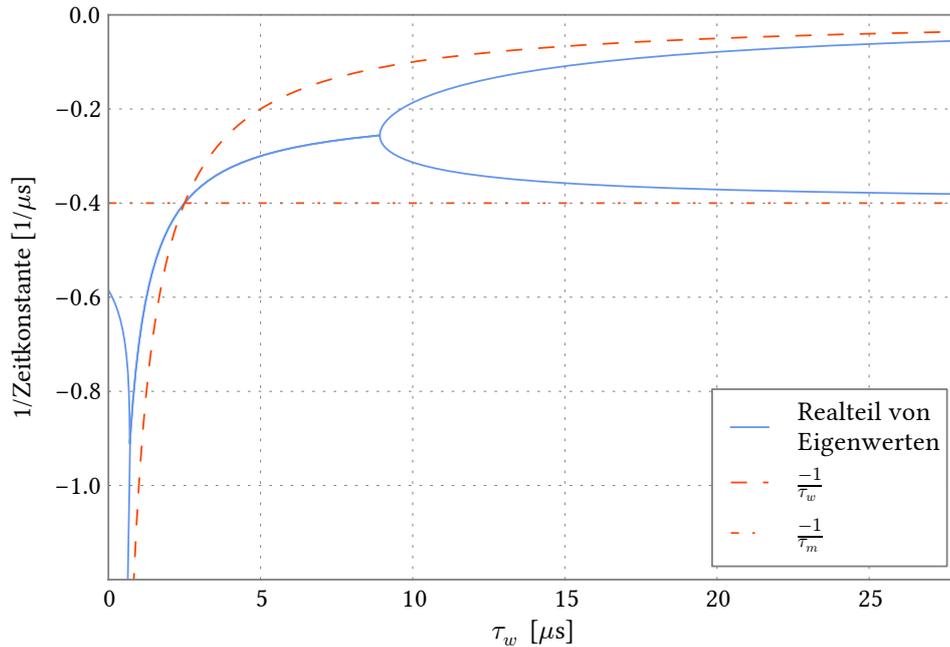


Abbildung 8: Realteil der Eigenwerte aus Gleichung 16 in Abhängigkeit von  $\tau_w$ . Hierbei sind:  $\tau_m = 2,5 \mu\text{s}$  und  $a = 0,4 \mu\text{s}$ . Die gestrichelten Linien zeigen die inversen Zeitkonstanten. Man erkennt mehrere Bifurkationspunkte. Daraus lässt sich schließen, dass zum Fitting verschiedener Hardware-Parameter sowohl die reelle als auch die komplexe Lösung der Membranspannung nötig sind. Für große  $\tau_w$  nähern sich die Eigenwerte jeweils an  $-1/\tau_w$  und  $-1/\tau_m$  an.

Die Nutzbarkeit dieser Lösungen für Hardwaremessungen muss nun evaluiert werden. Entscheidend ist hierbei das Verhalten von Gleichung 24 und 25 innerhalb von Parameterbereichen, die auf dem HICANN-Chip möglich sein können. Die im Design vorgegebenen Intervalle der relevanten Hardware-Parameter wurden daher aus *Millner (2012)* entnommen und in Tabelle 2 zusammengefasst.

Abbildung 8 zeigt für feste Werte von  $\tau_m$  und  $a$  die Abhängigkeit zwischen den Realteilen der Eigenwerte aus Gleichung 16 und  $\tau_w$ . Das dargestellte Intervall entspricht dabei nicht genau den Vorgaben aus Tabelle 2. Da jedoch noch nicht bekannt ist, ob die tatsächlichen Werte der Adaptionsparameter diese Vorgaben einhalten, wird hier und in folgenden Kapiteln  $\tau_w$  in einem Bereich von 0 bis  $60 \mu\text{s}$  untersucht. Werte oberhalb von  $28 \mu\text{s}$  zeigen in diesem Fall aber keine interessanten Effekte mehr, und wurden dementsprechend nicht dargestellt. Neben den angesprochenen Realteilen wurden zum Vergleich die Werte für  $-1/\tau_w$  und  $-1/\tau_m$  visualisiert. Im Verlauf der Eigenwerte sind mehrere Bifurkationspunkte der Realteile deutlich werden. Demnach kann nicht davon ausgegangen werden, dass innerhalb der vorstellbaren Bereiche der Hardware-Parameter die Lösung immer komplex oder reell ist. Aus diesem Grund wurde eine automatische Fallunterscheidung implementiert.

In Abbildung 9 wurden nun verschiedene simulierte Spannungsverläufe miteinander verglichen. Dabei ist jeweils die analytische und numerische Berechnung für  $\tau_w = 2 \mu\text{s}$

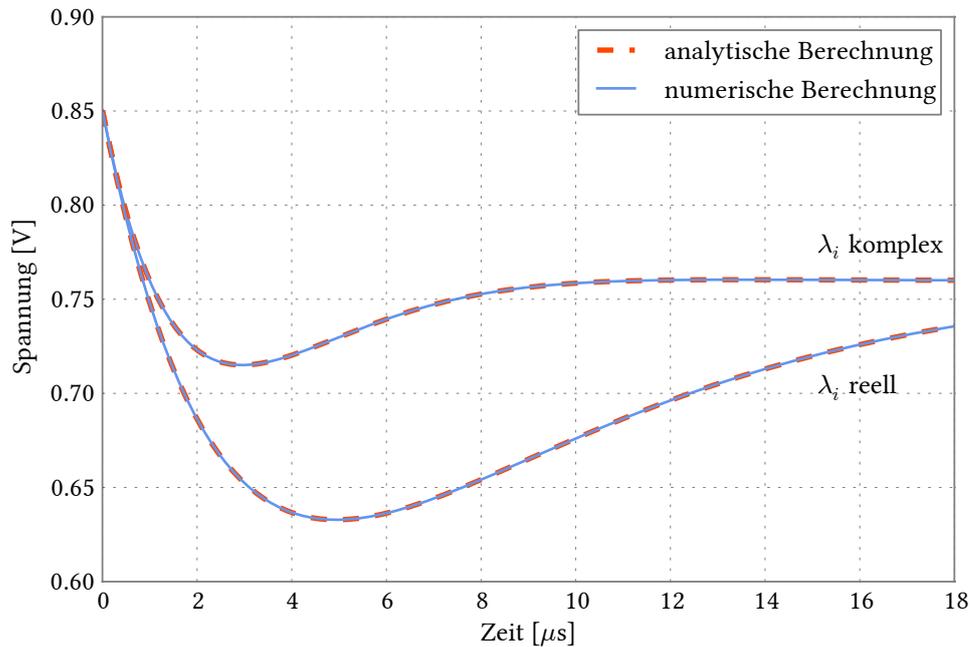


Abbildung 9: Vergleich von simulierten Spannungsverläufen, die sowohl mit der analytischen als auch der numerischen Lösung für definierte Parameter berechnet wurden. Alle Parameter außer  $\tau_w$  sind konstant und in Tabelle 13 zusammengefasst. Mit beiden Lösungen wurden für  $\tau_w = 2 \mu\text{s}$  und  $\tau_w = 10 \mu\text{s}$  Spannungsverläufe ermittelt, und im Verhältnis zur Zeit dargestellt. Nach Abbildung 8 entspricht das jeweils zwei komplexen und zwei reellen Eigenwerten. Man erkennt, dass die Verläufe das, in Kapitel 2.2 dargestellte, Verhalten von *subthreshold*-Adaption wieder spiegeln. Außerdem bestätigt sich, dass die analytische und numerische Lösung von Gleichung 12 im komplexen und reellen Fall äquivalent sind.

und  $\tau_w = 10 \mu\text{s}$  dargestellt. Die Werte von  $a$  und  $\tau_m$  entsprechen jenen aus Abbildung 8, wodurch jeweils ein Trace mit komplexen und reellen Eigenwerten zu sehen ist. Alle übrigen Parameter wurden konstant gehalten und sind in Tabelle 13 zusammengefasst. Die Überlagerung von numerischer und komplexer Berechnung deutet an, dass die beiden Lösungsmethoden äquivalent sind. Außerdem wird deutlich, dass die Funktionen die erwartete Form eines *subthreshold*-Spannungsverlaufs, wie in Abschnitt 2.2, durch eine auf dem HICANN-Chip vorstellbare Parameterkonfiguration wiedergeben kann. Eine Vergrößerung von  $\tau_w$  bewirkt zudem, wie in Abschnitt 2.2 angedeutet, eine Verlängerung der Überschwingung der Spannung unter das Leckpotential.

In den folgenden Abschnitten wurde zum Entwickeln und Testen der Fitroutine die numerische Lösung genutzt. Um die Dauer der Messungen und schlussendlich auch der Kalibration zu verkürzen, wäre es natürlich einer der ersten Schritte, die analytische Berechnung in die Methode einzupflegen. Bei einfachen Tests fällt die erhöhte Laufzeit jedoch noch nicht ins Gewicht.

	$\tau_m$ [ $\mu s$ ]	$\tau_w$ [ $\mu s$ ]	$a$ [ $\mu S$ ]
Min	0,54	10	0,4
Max	5,4	60	4

Tabelle 2: Im Design vorgegebene Bereiche von Hardware-Parametern, die für die Fitroutine relevant sind. Entnommen aus *Millner* (2012, S. 66).

### 3.3.2. Messprinzip

Es konnte bereits gezeigt werden, dass eine eindeutige Lösung von Gleichung 12 existiert. Im Folgenden wird daher eine Optimierungsroutine beschrieben, welche den im vorherigen Kapitel entwickelten Lösungsalgorithmus nutzt, um die Adaptionparameter  $\tau_w$  und  $a$  zu bestimmen. Wie bereits zu Beginn von Abschnitt 3.3 erwähnt, wird der Parameter  $b$  dabei nicht ermittelt.

Prinzipiell ähnelt die Vorgehensweise derjenigen zur Messung der Membranzeitkonstante aus Abschnitt 3.2.1. Die Membranspannung  $V_{mem}$  eines Neurons wird mithilfe eines externen Stromstimulus periodisch angeregt. Dadurch entstehen wiederkehrende Peaks im zeitlichen Verlauf von  $V_{mem}$ . Wenn die Periodendauer des Stromstimulus bekannt ist, kann ein so aufgenommener Analogtrace der angeregten Membran in Teile von der Länge einer Periode zerschnitten werden. Durch Mittelung dieser Traceteile entsteht ein Spannungsverlauf, der die Form der Membranspannung innerhalb einer Periodendauer des Stromstimulus mit stark reduziertem Rauschen wiedergibt. Wenn nun  $\tau_w, a > 0$  gilt, ist der Verlauf der abfallenden Flanke des dargestellten Peaks nach den Gleichungen 24 und 25 abhängig von den Parametern  $\tau_w$  und  $a$ . Aus diesem Grund wird mithilfe eines geeigneten Fitverfahrens, die Lösung aus Abschnitt 3.3.1 an die Membranspannung angenähert. Es ist daher wichtig den Spannungsverlauf der fallenden Flanke, das heißt, den Bereich in dem kein Stromstimulus auf die Membran wirkt, möglichst exakt bestimmen zu können. Hierfür wurde eine Methode zur Detektion von Kanten in einem Trace entwickelt. Es ist damit möglich das genaue Ende eines durch den Stromstimulus induzierten Rechteckpuls zu bestimmen. Dieser Zeitpunkt stellt den Anfang des Fitbereiches dar. Als Ende wird der Zeitpunkt  $t$  definiert, für den gilt:  $|V_{mem}(t) - E_l| > 3$  mV. Grund hierfür ist, dass alle Effekte der *subthreshold*-Adaption beobachtet werden sollen. Die Dauer dieser Effekte ist jedoch abhängig von der Größe der Adaptionzeitkonstante. Daher wäre es nicht praktikabel für alle Werte von  $\tau_w$  die Größe des Fitbereiches gleich zu wählen. Außerdem werden mit der beschriebenen Methode, bei einem kleineren  $\tau_w$ , keine unnötigen Traceabschnitte an das Optimierungsverfahren übergeben.

Für die eigentliche Fitroutine wird die Methode der kleinsten Quadrate verwendet. Das bedeutet, die Kostenfunktion wird unter Anwendung des Levenberg-Marquardt-Algorithmus minimiert (*Levenberg*, 1944). Umgesetzt wird dies mithilfe des Python Moduls *scipy.optimize* (*Jones et al.*, 2001).

Es muss bei dieser Fitmethode darauf geachtet werden, dass die Minimierung der Kostenfunktion auch lokale Minima als bestes Ergebnis betrachten kann, die nicht dem globalen Minimum entsprechen. Aus diesem Grund kann eine Abhängigkeit der ermittelten Parameter von der Startnäherung bestehen. Das ist vor Allem der Fall, wenn die Dimension des Parameterraums der Fitfunktion sehr hoch ist. Um die Stabilität der Fitmethode zu erhöhen, wird deswegen versucht möglichst viele Parameter aus Gleichung 24 vor Beginn der Regression zu ermitteln und diese im Fitverfahren als konstante Werte zu definieren.

Im Folgenden sind alle festen Parameter und deren Bestimmungsmethoden aufgelistet:

**Anfangsbedingung der Spannung**  $V_0$  ist als der erste Spannungswert des anzufittenden Traces definiert. Dabei muss darauf geachtet werden, dass  $V_0$  für die numerische und analytische Berechnung eine unterschiedliche Bedeutung haben. Die numerische Methode verlangt den exakten Wert der Spannung bei  $t = 0$ , wogegen die analytische Lösung die Differenz zwischen dieser Spannung und dem Leckpotential, das heißt, der partikulären Lösung, benötigt.

**Leckpotential:**  $E_l$  wird als der asymptotische Spannungswert für große Zeiten  $t$  des anzufittenden Traces definiert. Da die Membranspannung sich in jedem Trace nach endlicher Zeit auf einen konstanten Wert einstellt, sollte diese Definition den theoretischen Vorgaben des AdEx-Modells am ehesten entsprechen.

**Membrankapazität** Die Bestimmung von  $C_m$  stellt eine großes Problem für die Genauigkeit der Messmethode dar. Es existierte zum Zeitpunkt der Abgabe dieser Arbeit nur die Möglichkeit die Membrankapazität in Abhängigkeit vom externen Stromstimulus zu bestimmen (Millner, 2012). Diese Stromstärke kann jedoch nicht exakt quantifiziert werden. Daher muss als Richtwert für die Kapazität eine Abschätzung aus Millner (2012) von  $C_m = 3,8$  pF verwendet werden. Der Vorteil von dieser Schätzung gegenüber dem nominell eingestellten Wert liegt darin, dass die parasitäre Kapazität des Stromstimulus bereits berücksichtigt wurde. Nichtsdestotrotz werden so produktionsbedingte Unterschiede der Kapazitäten vernachlässigt. Es wäre demnach sinnvoll  $C_m$  als freien Parameter an die Fitroutine zu übergeben. Wie sich in Abschnitt 3.3.3 jedoch herausstellen wird, ist es im Moment wichtiger die Dimensionen der Fitfunktion zu minimieren, da sonst zu viele lokale Minima im Parameterraum existieren. Im Gegensatz zu der im vorherigen Kapitel behandelten Messmethode besteht nun aber zumindest während der Bestimmung von allen Parametern derselbe systematische Fehler von  $C_m$ . In der Messmethode aus Abschnitt 3.2.1 war das nicht der Fall. Hier wurde die Spikeperiode, sowie  $V_{\text{reset}}$  und  $V_{\text{th}}$ , ohne den Einfluss der parasitären Kapazität vom externen Stromstimulus bestimmt. Wenn zukünftige Versionen der Fitroutine das globale Minimum zuverlässiger bestimmen können, sollte jedoch für ein genaueres Ergebnis, und gleichzeitig für eine Ermittlung der effektiven Kapazität,  $C_m$  als freier Parameter definiert werden.

**Membranzeitkonstante**  $\tau_m$  wird hier mit der gleichen Methode wie schon in Abschnitt 3.2.1 bestimmt. Durch Festlegung der Membranzeitkonstante für die Fitroutine, nimmt man in Kauf, dass *trial-to-trial*-Variationen beim schreiben der *floating-gates* nicht mehr ausgeglichen werden können. Dieses Problem kann man kompensieren, indem der Mittelwert über mehrere Messungen von  $\tau_m$  berechnet wird.

Damit müssen also noch die drei Parameter:  $\tau_w$ ,  $a$  und  $w_0$  von der Fitroutine ermittelt werden. Da wir nur an den Werten von  $\tau_w$  und  $a$  interessiert sind, würde man im optimalen Fall auch  $w_0$  festlegen. Jedoch gibt es auf dem HICANN-Chip keine Möglichkeit die Stromstärke  $w$  auszumessen (HBP SP9 partners, 2014). Deshalb wird in den folgenden Kapiteln, aus Ermangelung einer besseren Schätzung, die Startnäherung von  $w_0$  auf  $0,2$   $\mu\text{A}$  eingestellt und der Parameter mitbestimmt.

	$\tau_m$ [ $\mu s$ ]	$\tau_w$ [ $\mu s$ ]	$a$ [ $\mu S$ ]
Min	0	0	0
Max	5,4	60	5

Tabelle 3: Bereiche von Hardware-Parametern, für welche die Genauigkeit der in Abschnitt 3.3.2 beschriebenen Fitroutine untersucht wurde.

An dieser Stelle wird folgende Notation eingeführt, für einen Parameter  $X$  heißt die Startnäherungen  $X_S$ . Da noch unbekannt ist, in welchen Bereichen die tatsächlichen Werte von  $a$  und  $\tau_w$  auf dem HICANN-Chip liegen, kann für diese  $X_S$  nur eine grobe Schätzung anhand der per Design vorgegebenen Parameterbereiche (siehe Tabelle 2) abgegeben werden. Deshalb wird  $\tau_{w,S}$  und  $a_S$  auf etwa die Mitte des nominell nutzbaren Bereiches gestellt. Dies erhöht jedoch die Möglichkeit, dass die Fitmethode sich einem falschen lokalen Minimum annähert. Um erste Beziehungen zwischen den Adaptionparametern und ihren jeweiligen einstellbaren *floating-gate*-Spannungen herzustellen, wird daher jeweils nur über eine der beiden Kontrollspannungen iteriert. Auf diese Weise kann vor der Messung anhand einer Simulation bestimmt werden, für welche Bereiche des nicht iterierten Parameters der Fit eine hohe Genauigkeit aufweist.

Im folgenden Abschnitt wird eine Simulationsmethode vorgestellt, mit der solche Bereiche gefunden werden können.

### 3.3.3. Validierung

Neben der bereits erwähnten Ermittlung von geeigneten Bereichen des jeweils nicht iterierten Adaptionparametern, wird mithilfe der hier beschriebenen Simulationsebene auch die Abhängigkeit der Fitgenauigkeit von der Membranzeitkonstante untersucht.

Prinzipiell werden dabei Spannungsverläufe durch die numerische Lösung des vereinfachten AdEx-Modells aus Gleichung 12 berechnet. Es können so für definierte Werte der Parameter  $\tau_m$ ,  $\tau_w$  und  $a$  Traces simuliert werden. Man erhält dadurch Referenzwerte für die gemessenen Parameter der Fitroutine und kann deren Genauigkeit abschätzen. Die Güte des gefundenen Parameters ist durch die relative Abweichung von gemessenen und vorgegebenen Wert quantifiziert.

Des Weiteren wird ein schwaches Rauschen emuliert, indem auf die Spannungsverläufe Stichproben aus einer gaussischen Verteilung mit einer Breite von  $\sigma = 0,3$  mV addiert werden.

$K_S$  des nicht iterierten Parameters  $K$  wird zufällig aus in einem Intervall von  $\pm 80\%$  um den, in der Simulation vorgegebenen, Wert von  $K$  ausgesucht. Hierdurch werden produktionsbedingte Schwankungen des konstant gehaltenen Parameters in der Simulation berücksichtigt. Natürlich ist die Größe von  $K$  bei einer Hardwaremessungen nicht bekannt. Hier wird versucht durch Einzelmessungen in verschiedenen Bereichen des iterierten Parameters eine erste Schätzung für  $K$  zu erhalten und damit  $K_S$  zu definieren.

Die untersuchten Intervalle von  $\tau_m$ ,  $\tau_w$  und  $a$  sind in Tabelle 3 zusammengefasst. Sie orientieren sich an der theoretischen Parameterbereichen aus Tabelle 2. Sie sind jedoch leicht vergrößert, um die Fitmethode auch für abweichende Werte der Adaptionparameter untersuchen zu können.

#### Simulation der Messung von $a$

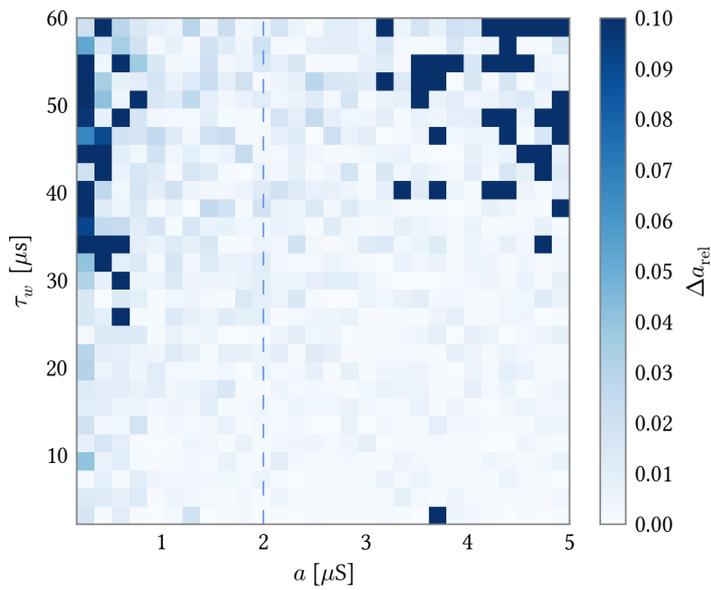


Abbildung 10: Plot des relativen Fehlers der Fitroutine von  $a$  in Abhängigkeit von der  $\tau_m$  und  $a$ . Es wurden dabei Spannungsverläufe für den dargestellten Bereich von  $\tau_m$  und  $a$  simuliert und mithilfe der numerischen Lösung des AdEx-Modells ohne exponentiellen Term eine Parameteroptimierung durchgeführt. Die gestrichelte Linie stellt die Startnäherung der Fitroutine dar.  $\tau_m$  ist konstant auf  $2,5 \mu\text{s}$  gestellt. Alle Datenpunkte die den Maximalwert von  $\Delta a_{\text{rel}}$  erreichen, liegen mindestens um einen Faktor von  $10^3$  darüber und wurden zur besseren Darstellung begrenzt.

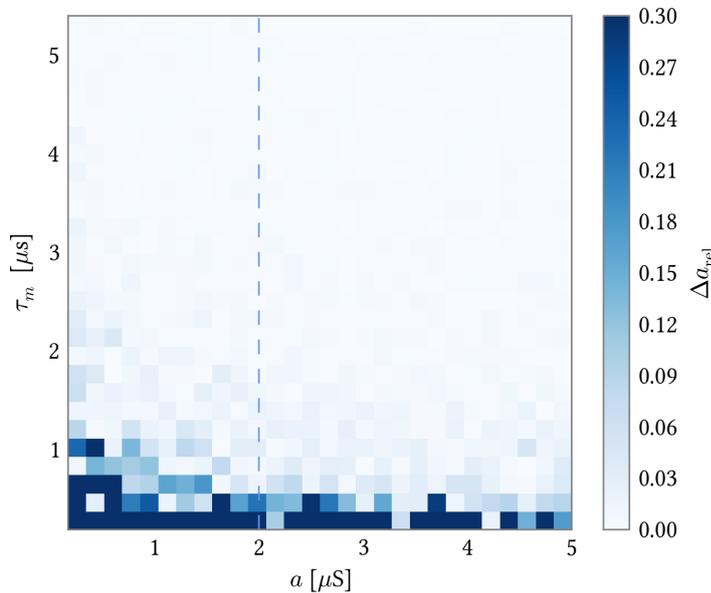


Abbildung 11: Dieser Plot zeigt die Beziehung von  $\Delta a_{\text{rel}}$  zu  $\tau_m$  und  $a$ . Das Darstellungsprinzip entspricht dem von Abbildung 10.  $\tau_w$  liegt konstant bei  $10 \mu\text{s}$  und die Startnäherung (gestrichelte Linie) bei  $2 \mu\text{s}$ .

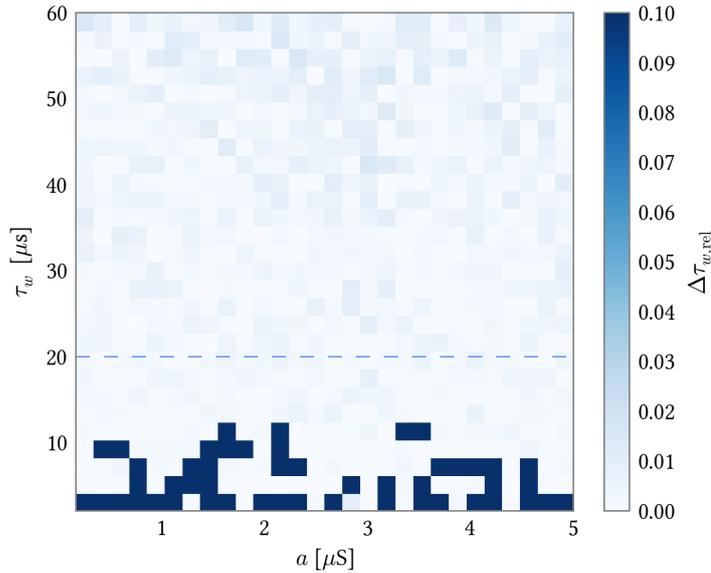


Abbildung 12: Hier wird der relative Fehler der Fitroutine von  $\tau_w$  in Abhängigkeit von  $\tau_w$  und  $a$  dargestellt. Das Simulationsprinzip ist ähnlich wie in Abbildung 10. Für die Membranzeitkonstante gilt:  $\tau_m = 2,5 \mu\text{s}$  und für die Startnäherung:  $\tau_{w,S} = 20 \mu\text{s}$

Der relative Fehler  $\Delta a_{\text{rel}}$  von  $a$  wird nun für verschiedene Punkte aus dem Parameterraum  $(\tau_m, \tau_w, a)$  dargestellt. In Abbildung 10 sind die Ergebnisse für eine feste Membranzeitkonstante von  $\tau_m = 2,5 \mu\text{s}$  in Abhängigkeit von  $a$  und  $\tau_w$  geplottet. Die gestichelte Linie zeigt die verwendete Startnäherung  $a_S$  an. Der maximal dargestellte Fehler von  $a$  wurde hier auf 0,1 begrenzt.

Generell wird auch in den folgenden Diagrammen der gezeigte Maximalwert von  $\Delta X_{\text{rel}}$  begrenzt sein. Der tatsächliche Wert aller Datenpunkte, die dieses Maximum erreichen, ist mindestens um einen Faktor von  $10^3$  größer. Daraus kann geschlossen werden, dass für diese Parameter ein falsches lokales Minimum gefunden wurde.

Man erkennt, dass die Fitroutine für alle Datenpunkte in der Nähe der vorgegebenen Startnäherung den Parameter  $a$  präzise bestimmen kann. Bei einer größeren Differenz von  $a$  und  $a_S$  konvergiert die Fitfunktion gehäuft in ein falsches lokales Minimum. Das war zu erwarten und unterstreicht nochmal die Bedeutung von einer sinnvollen Startnäherung. Eine mögliche Erklärung für die allgemein schlechteren Ergebnisse bei großem  $\tau_w$  kann mithilfe des Bifurkationsdiagramms (siehe Abbildung 8) aus Abschnitt 3.3.1 gefunden werden. Darin erkennt man, dass bei langen Adaptionszeitkonstanten die Eigenwerte  $\lambda_i$  der Lösung des vereinfachten AdEx-Modells reell werden. Gleichzeitig gilt für große  $\tau_w$ :

$$\lambda_1 \rightarrow -\frac{1}{\tau_m} \quad \text{und} \quad \lambda_2 \rightarrow -\frac{1}{\tau_w} .$$

Damit nähert sich also  $\lambda_2$  für steigende  $\tau_w$  an 0 an. Wenn wir außerdem annehmen, dass die Eigenvektoren kein gegenteiliges Verhalten zeigen, so kann die Lösung  $V(t)$  des AdEx-Modells ohne exponentiellen Term zunehmend gut durch eine Exponentialfunktion mit nur einer Zeitkonstante beschrieben werden. Die hier genutzte Fitfunktion versucht je-

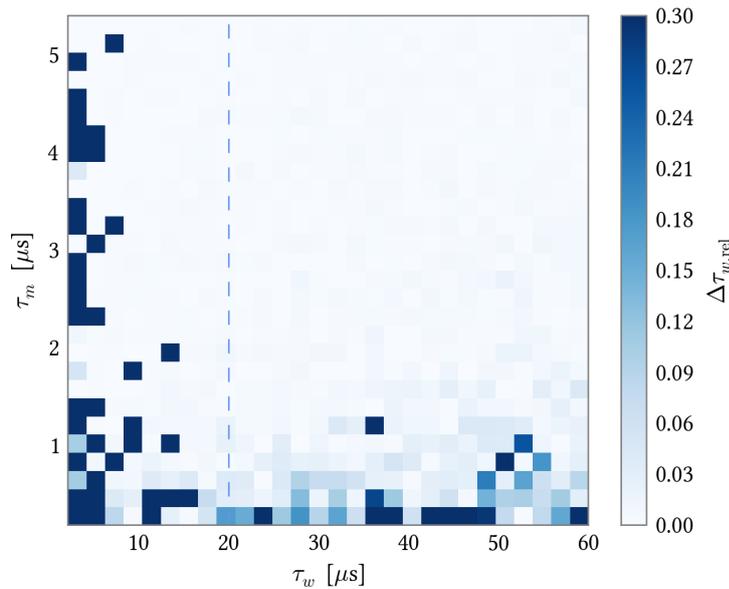


Abbildung 13: Ähnlich wie in Abbildung 11 wird hier die Abhängigkeit der Fitroutine für  $\tau_w$  von  $\tau_m$  untersucht.  $a$  liegt dabei konstant auf  $4 \mu\text{s}$ .

doch effektiv die Werte von  $\lambda_1$  und  $\lambda_2$ , das heißt, von zwei verschiedenen Zeitkonstanten, zu optimieren. Man erhält dadurch eine überbestimmte Kostenfunktion. Wenn das der Fall ist, kann der Fit-Algorithmus vor allem bei ungenauen Startnäherungen leicht in eine falsche Richtung konvergieren. Um diesen Erklärungsansatz zu validieren, müsste noch das Verhalten der Eigenvektoren  $v_i$  genauer betrachtet werden. Für Hardwaremessungen sollte, bei einer Wahl von geeigneten Werten für  $\tau_m$  und  $\tau_w$ , dieser Effekt nur noch eine untergeordnete Rolle spielen und wird daher nicht tiefer untersucht.

Es muss nun also ein Bereich für  $\tau_w$  gefunden werden, in dem  $\Delta a_{\text{rel}}$  für alle Werte von  $a$  möglichst klein ist. Wie in Abbildung 10 deutlich wird, ist das am ehesten für kleine Adaptionszeitkonstanten der Fall. Bei der Messung von  $a$  auf dem HICANN-Chip sollte demnach ein möglichst kleines  $\tau_w$  eingestellt werden.

In der nächsten Simulation wird deshalb die Adaptionszeitkonstante auf den nach Vorgaben aus Tabelle 2 kleinstmöglichen Wert von  $\tau_w = 10 \mu\text{s}$  gestellt. Nun kann nach dem gleichen Prinzip wie in Abbildung 10 die Beziehung von  $\Delta a_{\text{rel}}$  zu  $\tau_m$  und  $a$  ermittelt werden.

Abbildung 11 zeigt diese Ergebnisse für die in Tabelle 3 zusammengefassten Bereiche. Hier wird deutlich, dass eine klare Abhängigkeit der Fitgenauigkeit von der Membranzeitkonstante existiert. Für große Werte von  $\tau_m$  kann in der Simulation, bei ausreichend kleinem  $\tau_w$ , der Parameter  $a$  über seinen gesamten Bereich mit einer konstanten Startnäherung bestimmt werden.

Aus den gezeigten Simulationen folgt also, dass für Hardwaremessungen von  $a$  eine hohe Membran- und eine niedrige Adaptions-Zeitkonstante eingestellt werden sollte.

#### Simulation der Messung von $\tau_w$

Nach dem gleichen Prinzip wie zur Bestimmung von  $\Delta a_{\text{rel}}$  wird nun der relative Fehler der Adaptionzeitkonstante in Abhängigkeit von  $\tau_m$ ,  $\tau_w$  und  $a$  dargestellt.

Dazu ist in Abbildung 12 erneut für eine konstante Membranzeitkonstante von  $\tau_m = 2,5 \mu\text{s}$  die Beziehung des relative Fehlers zu  $\tau_w$  und  $a$  aufgetragen.  $\tau_{w,S}$  wurde auf  $20 \mu\text{s}$  gesetzt. Für Adaptionzeitkonstanten im Bereich von 0 bis  $10 \mu\text{s}$  erreicht die Optimierungsmethode für den gesamten Intervall von  $a$  nicht das korrekte lokale Minimum. Oberhalb von  $10 \mu\text{s}$  ermittelt die Fitroutine das globale Minimum mit ausreichender Genauigkeit. Dabei ist keine eindeutige Abhängigkeit vom Parameter  $a$  zu erkennen. Für Hardwaremessungen wird dennoch ein hoher Wert von  $a$  eingestellt, um den Einfluss der Adaptionseffekte auf den Spannungsverlauf zu vergrößern.

Die folgende Simulation nutzt deswegen einen hohen Adaptionfaktor von  $a = 4 \mu\text{S}$ . In Abbildung 13 ist ähnlich wie im vorherigen Abschnitt nun  $\Delta\tau_{w,\text{rel}}$  in Abhängigkeit von  $\tau_m$  und  $\tau_w$  dargestellt. Wie schon in Abbildung 11 vergrößert sich der relative Fehler für kleinere Membranzeitkonstanten. Jedoch werden Werte von  $\tau_w$  zwischen 0 und  $10 \mu\text{s}$  auch hier nicht korrekt ermittelt. Man muss demnach damit rechnen, dass die Fitmethode bei der Bestimmung von sehr kleinen Zeitkonstanten auf der Hardware instabil wird. Für den Intervall von 10 bis  $60 \mu\text{s}$  kann der Fitalgorithmus bei großem  $\tau_m$  dennoch präzise Werte von  $\tau_w$  ermitteln.

Insgesamt sollte also für die Messung der Adaptionzeitkonstante auf dem HICANN-Chip sowohl der Adaptionfaktor als auch die Membranzeitkonstante möglichst groß sein.

## 4. Resultate

### 4.1. Messung von $\tau_m$

Die Bestimmung der Membranzeitkonstante wird, wie in Abschnitt 3.3.2 angedeutet, für die Messung der Adaptionparameter benötigt.  $\tau_m$  wurde dabei durch das in Kapitel 3.2.1 beschriebene Messprinzip ermittelt. Für jedes Neuron wurden 3 Messungen durchgeführt. In Abbildung 14 ist die Verteilung von der Mittelwerte von  $\tau_m$  aller Neuronen für einen Biasstrom von nominell  $I_{\text{gl}} = 489 \text{ nA}$  dargestellt. Das entspricht einen eingestellten DAC-wert von  $I_{\text{gl}} = 200$ . Wie in Abbildung 11 gezeigt wurde, würde man im optimalen Fall ein maximales  $\tau_w$  durch  $I_{\text{gl}} = 0$  DAC einstellen. Bei einer solchen Konfiguration waren die Membranzeitkonstanten von vielen Neuronen jedoch sehr groß. Dadurch konnte der freie Zerfall der Membranspannung innerhalb von einer Periode des Stromstimulus nicht mehr vollständig dargestellt werden. Für den Biasstrom wurde deshalb ein Wert gewählt bei dem die Membranzeitkonstante noch für alle Neuronen bestimmt werden konnte.

In *Schmidt (2014, S. 41)* ist die Abhängigkeit der Membranzeitkonstanten von dem Biasstrom  $I_{\text{gl}}$  näher untersucht worden. Man erkennt hierbei einen nichtlinearen Zusammenhang. Eine symmetrische Streuung von  $I_{\text{gl}}$  durch *floating-gate*-Variationen würde sich demnach in einer asymmetrischen Verteilung der Membranzeitkonstanten bemerkbar machen. Eine solche Distribution ist in Abbildung 14 sichtbar.

Der Mittelwert über allen Neuronen beträgt:  $\bar{\tau}_m = 2,21 \mu\text{s}$ . Dieses Ergebnis liegt in einem Bereich, der schon in einer vorherigen Arbeit ermittelt wurde (*Schmidt, 2014, S. 45*). Die gemessenen Membranzeitkonstanten werden deshalb für die Fitroutine von  $a$  und  $\tau_w$  verwendet.

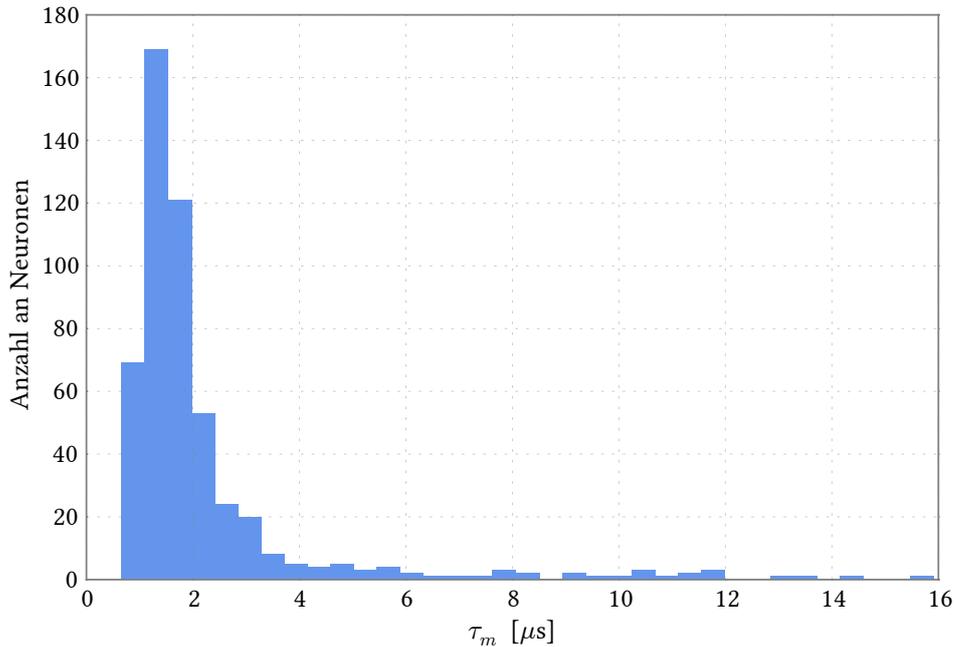


Abbildung 14: Verteilung der Membranzeitkonstanten für  $I_{gl} = 200$  DAC über dem gesamten HICANN-Chip. Aufgenommen auf dem Vertical Setup *porthos*.

Mit der gewählten Konfiguration liegt  $\tau_m$  von ca. 71,1 % der Neuronen jedoch unter 2  $\mu\text{s}$ . Wie in Abbildung 11 und 13 zu sehen ist, nimmt die Genauigkeit der Fitroutine in diesem Bereich ab. Bei der Betrachtung der folgenden Ergebnisse muss dieser Umstand berücksichtigt werden.

#### 4.2. Messung von $a$

Die in Abschnitt 3.3.2 beschriebene Messmethode der Konduktanz  $a$  wird nun auf dem HICANN-Chip angewendet. Die Adaptionszeitkonstante wird, durch  $I_{\text{radapt}} = 1023$  DAC, auf den minimal möglichen Wert eingestellt. Die Dauer eines Rechteckpulses des Stromstimulus beträgt 17,6  $\mu\text{s}$  mit einer Stärke von 90 DAC. Der gesamte verwendete Parametersatz ist in Tabelle 12 zusammengefasst.

Die Startnäherungen der Fitroutine lauten hierbei:

$$\tau_{w,S} = 5 \mu\text{s} , a_S = 2 \mu\text{S} , w_{0,S} = 0.1 \mu\text{A} .$$

Es fällt auf, dass  $\tau_{w,S}$  kleiner ist als das im Design vorgegebene Intervall (siehe Tabelle 2). Die Messungen in Abschnitt 4.3 zeigen jedoch, dass das Designintervall von  $\tau_w$  mit hoher Sicherheit nicht dem tatsächlichen Parameterbereich auf dem HICANN-Chip entspricht. Um eine genauere Bestimmung von  $a$  zu ermöglichen wurde deshalb eine niedrigere Startnäherung für  $\tau_w$  gewählt.

Mit diesen Startnäherungen wurden in Abbildung 15 für verschiedene Werte von  $I_{\text{gladapt}}$  die gemittelten Spannungsverläufe und die dazugehörigen Fitkurven dargestellt. Man erkennt, dass die durch den Fitalgorithmus ermittelten Werte die gemessenen Spannungs-

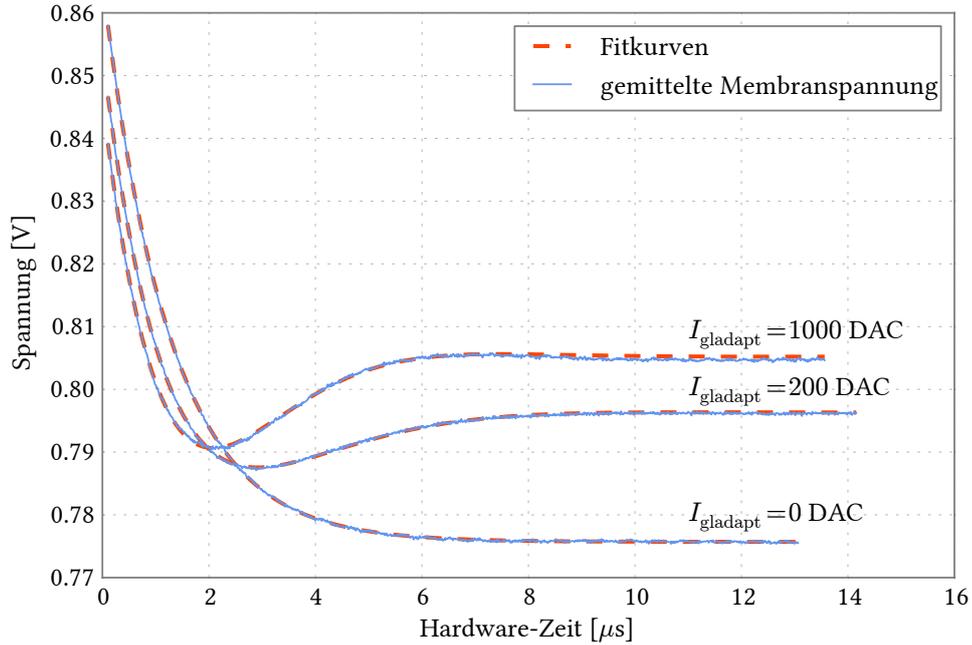


Abbildung 15: Beispiel eines Fits zur Bestimmung des Parameters  $a$  für 3 verschiedene  $I_{\text{gladapt}}$ . Die ermittelten Werte sind in Tabelle 4 zusammengefasst. Die gewählte Hardwarekonfiguration ist in Tabelle 12 aufgelistet.

$I_{\text{gladapt}}$	$\tau_w$ [ $\mu\text{s}$ ]	$a$ [ $\mu\text{S}$ ]	$w_0$ [nA]
0	122,8	-6,04	5,0
200	2,06	1,34	97,0
1000	1,83	2,46	149,1

Tabelle 4: Fitergebnisse der 3 verschiedenen Fitkurven aus Abbildung 15.

verläufe mit ausreichender Genauigkeit wiedergeben können. Wie schon in Abschnitt 3.3.2 angemerkt, besteht jedoch eher die Frage, ob die Fitergebnisse den tatsächlichen Hardwareparametern entsprechen oder die Fitfunktion zu falschen lokalen Minima konvergiert ist. In Tabelle 4 sind daher die Fitparameter der 3 exemplarischen Spannungsverläufe zusammengefasst. Die Werte für  $I_{\text{gladapt}} = 0$  DAC machen deutlich, dass aus einer guten Übereinstimmung der Fits mit den Messdaten nicht zwingenderweise präzise Parametermessungen folgen. Für gegen 0 gehende Werte von  $a$  war dies auch zu erwarten. Das hier dargestellte Neuron hat eine Membranzeitkonstante von  $\tau_m = (1,287 \pm 0,006) \mu\text{s}$ . Abbildung 10 und 11 zeigen, dass in diesem Bereich bei kleinem  $a$  die Fitstabilität nachlässt. Die Messergebnisse für kleine  $I_{\text{gladapt}}$  werden in den folgenden Abbildungen demnach eine geringe Aussagekraft haben.

In Abbildung 16 ist nun der Adaptionsfaktor  $a$  und die Adaptionszeitkonstante  $\tau_w$  in Abhängigkeit von  $I_{\text{gladapt}}$  für ein einzelnes Neuron dargestellt. Es ist deutlich zu sehen, dass der Parameter  $a$  im Intervall von  $0,41 \mu\text{S}$  bis  $5,10 \mu\text{S}$  abhängig von seinem Biasstrom

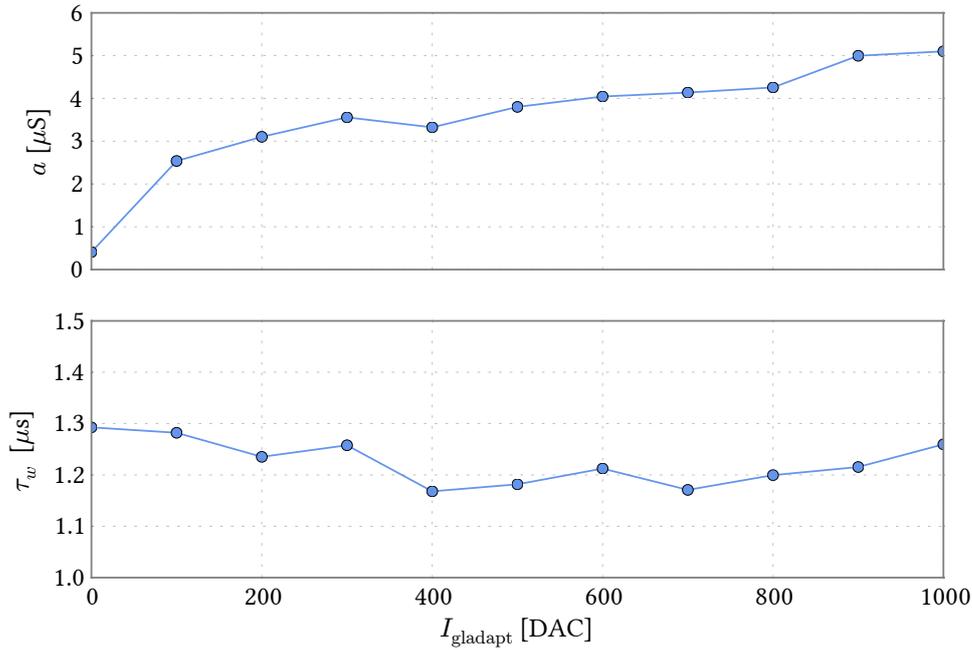


Abbildung 16: Messung des Adaptionsfaktors  $a$  über  $I_{\text{gladapt}}$  (oben), und der durch die Fitroutine gleichzeitig bestimmte Adaptionszeitkonstante  $\tau_w$  (unten). Dargestellt ist das Neuron 511 auf dem Vertical Setup *porthos* mit einer Membranzeitkonstante von  $\tau_m = (2,78 \pm 0,06) \mu\text{s}$ .

ansteigt. Gleichzeitig bewegt sich  $\tau_w$  nur zwischen  $1,17 \mu\text{s}$  und  $1,29 \mu\text{s}$  ohne erkennbaren Bezug zu  $I_{\text{gladapt}}$ .

Der erreichbare Intervall von  $a$  entspricht zu hohem Maße den Designvorgaben aus Tabelle 2. Auch das annähernd konstante Verhalten von  $\tau_w$  erfüllt die Erwartungen, da die Adaptionszeitkonstante durch den Biasstrom theoretisch nicht beeinflusst werden kann. Die Membranzeitkonstante des hier dargestellten Neurons liegt bei:  $\tau_m = (2,78 \pm 0,06) \mu\text{s}$ . In Abschnitt 3.3.3 wurde zudem gezeigt, dass die Abweichung der Ergebnisse von  $a$  und  $\tau_w$  mit steigender Membranzeitkonstante abnehmen. Dies könnte ein Indiz dafür sein, dass die hier gezeigten Messergebnisse eine hohe Genauigkeit ausweisen und die Fitfunktion nicht zu falschen Parametersätzen hin konvergiert. Interessanterweise liegt die Adaptionszeitkonstante bei durchschnittlich:  $\tau_w = (1,23 \pm 0,04) \mu\text{s}$ . Mit  $I_{\text{radapt}} = 1023 \text{ DAC}$  wurde  $\tau_w$  zwar auf den minimal möglichen Wert gestellt, jedoch soll dieser nach Designvorgaben  $10 \mu\text{s}$  betragen. Es wäre also möglich, dass die erwarteten Werte der Adaptionszeitkonstanten aus Tabelle 2 nicht erreicht werden.

Die Messung von  $a$  über  $I_{\text{gladapt}}$  wurde mit allen Neuronen auf dem Vertical Setup *porthos* durchgeführt. Abbildung 17 zeigt den Mittelwert dieser Messungen. Die Fehlerbalken stellen die Standardabweichung der Neuronen dar. Messwerte die unterhalb von 0 oder oberhalb von  $10 \mu\text{s}$  liegen wurden nicht dargestellt. Die relativen Anteile der dadurch verworfenen Messwerte zur Gesamtzahl an Neuronen sind in Tabelle 5 zusammengefasst. Darin wird deutlich, dass für einen Biasstrom von  $I_{\text{gladapt}} = 0 \text{ DAC}$  die Fitmethode sehr

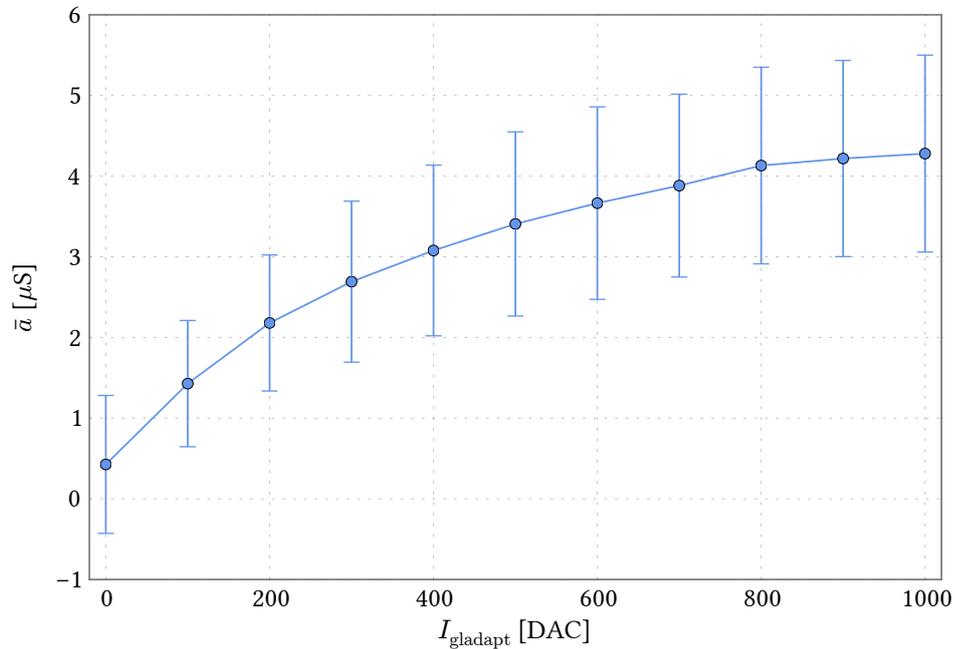


Abbildung 17: Mittelung der Messergebnisse von  $a$  über allen Neuronen eines HICANN-Chips in Abhängigkeit vom Biasstrom  $I_{\text{gladapt}}$ . Die Fehlerbalken stellen die Streuung der Neuronen um ihren Mittelwert dar. Werte, die unterhalb von 0 und oberhalb von  $10 \mu\text{S}$  liegen, wurden in der Berechnung von Mittelwert und Standardabweichung nicht berücksichtigt. Die relative Anzahl an dadurch verworfenen Neuronen pro Biasstromwert ist in Tabelle 5 zusammengefasst.

$I_{\text{gladapt}}$ [ $10^2$ DAC]	0	1	2	3	4	5	6	7	8	9	10
verworfenen $a$ [%]	32,4	6,6	2,7	1,1	0,7	0,2	1,2	1,0	1,4	0,6	1,4

Tabelle 5: Die relative Anzahl der in Abbildung 17 nicht beachteten Werte der Konduktanz  $a$ . In Abbildung 17 wurden nur Messwerte von  $a$  berücksichtigt, die in einem Intervall von 0 bis  $10 \mu\text{S}$  liegen. Messpunkte außerhalb dieser Grenzen, stellen entweder eine fehlgeschlagene Fitroutine oder Neuronen dar, die nicht auf den Mittelwert aller Neuronen kalibriert werden können. In beiden Fällen können diese Messpunkte im Hinblick auf eine Kalibration nicht genutzt werden.

$I_{\text{radapt}}$ [DAC]	50	150	250	350	450	550	650	750	850	950
$\bar{\tau}_w$ [ $\mu\text{s}$ ]	19,4	12,4	7,9	6,1	4,9	3,8	3,2	2,8	2,4	2,3
verworfenen $\tau_w$ [%]	38,7	16,0	8,5	7,4	8,4	14,5	20,0	29,5	38,3	49,0

Tabelle 6: Die Tabelle stellt den Mittelwert von Messungen der Adaptionzeitkonstante aller Neuronen dar. Hierbei wurde eine konstante Startnäherung von  $\tau_{w,s} = 20 \mu\text{s}$  und  $a_s = 3 \mu\text{s}$  verwendet. Außerdem ist der relative Anteil an Messwerten von  $\tau_w$ , die unterhalb von 0 und oberhalb von  $100 \mu\text{s}$  liegen, aufgelistet. Diese Werte wurden nicht für die Mittelwertberechnung verwendet.

große Schwankungen verursacht. Es besteht zwar auch die Möglichkeit, dass zumindest Messwerte oberhalb von  $10 \mu\text{s}$  einen systematischen Fehler der Hardware darstellen, jedoch wurde in Tabelle 4 bereits gezeigt, dass für gegen 0 gehende Werte des Biasstroms sich die Unsicherheit der Fitroutine vergrößert. Außerdem haben die Membranspannungen bei gegen 0 gehenden  $a$ , und bei sehr großem  $\tau_w$  sehr ähnliche Verläufe. Das heißt, ein derartig geformter Trace kann von mindestens 2 Parametersätzen gut beschrieben werden. Diese Problem wird auch noch bei der Bestimmung von großen Adaptionzeitkonstanten eine Rolle spielen.

Die Form der in Abbildung 17 dargestellten Kurve ähnelt einer Wurzelfunktion. In Millner (2012, S. 37) wurde erklärt, dass für die Abhängigkeit der zur Implementierung der Adaption genutzten OTAs von ihrem Biasstrom gelten soll:  $g_x \sim \sqrt{I_x}$ , wobei  $g_x$  die Konduktanz von  $\text{OTA}_x$  mit einem Biasstrom  $I_x$  ist. Demnach stellt die Messung von  $a$  die gleichen Beziehung zum Biasstrom dar, wie sie im Design vorgegeben ist. Hierzu wurde in Anhang A eine Gegenüberstellung der aufgenommenen Messdaten mit einer Transistorlevel-Simulation aus Millner (2012, S. 39) durchgeführt. In dieser ist die Vergleichbarkeit der Messpunkte mit den Simulationsdaten deutlich sichtbar.

Neben der Form der aufgenommenen Messkurve wird deutlich, dass der ermittelte Intervall von  $a$  weitgehend den Designvorgaben aus Tabelle 2 entspricht. In Abbildung 18 ist hierzu die Verteilung von  $a$  für einen konstanten Biasstrom von  $I_{\text{gladapt}} = 500 \text{ DAC}$  gezeigt. Diese kann als annähernd symmetrisch beschrieben werden. Im Hinblick auf eine Kalibrierung des Adaptionsverhaltens sollte es demnach möglich sein, den Adaptionsfaktor von einem Großteil der Neuronen auf den im Design vorgegebenen dynamischen Bereich einzustellen.

### 4.3. Messung von $\tau_w$

Bei der Bestimmung der Adaptionzeitkonstante geht man ähnlich vor wie schon zur Analyse des Adaptionsfaktors. Jedoch muss die Messroutine in den folgenden drei Punkten angepasst werden.

**gewählter Adaptionsfaktor** In Abschnitt 3.3.3 wurde erklärt, dass für eine genaue Messung von  $\tau_w$  ein großes  $a$  vorteilhaft ist. Es hat sich bestätigt, dass sehr kleine Werte des Adaptionsfaktors die Messunsicherheit vergrößern. Jedoch ist die Übereinstimmung von Fitkurve und Membranspannung in Abbildung 15 für ein maximales  $I_{\text{gladapt}} = 1000 \text{ DAC}$  schlechter als bei dem mittleren Wert von  $a$ . Außerdem zeigt Tabelle 5, dass für  $I_{\text{gladapt}} = 500 \text{ DAC}$  die Anzahl an verworfenen Messwerten am geringsten ist. Aus diesem Grund wurde für die Untersuchung von  $\tau_w$  der Biasstrom

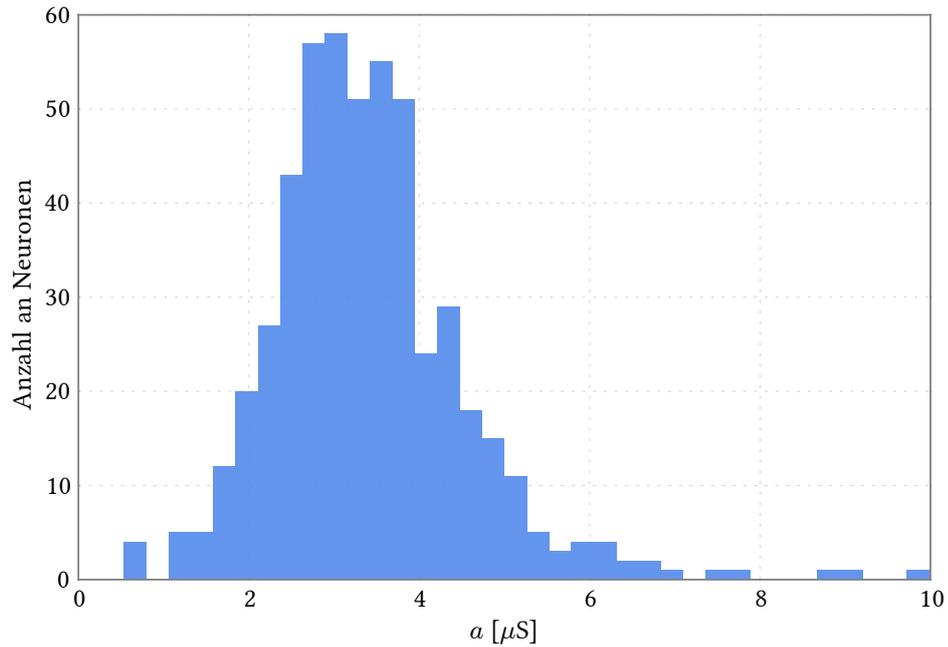


Abbildung 18: Verteilung des Adaptionfaktors  $a$  für einen festen Biasstrom von  $I_{\text{gladapt}} = 500$  DAC auf dem Vertical Setup *porthos*. Der relative Anteil an Messpunkten, die unterhalb von 0 oder oberhalb von  $10 \mu\text{S}$  liegen, ist 0,2 %.

$I_{\text{gladapt}}$  von 500 DAC gewählt.

**Messbereich** Wie im vorherigen Abschnitt bereits beschrieben wurde, ist die Bestimmung von sehr großen Adaptionzeitkonstanten nur mit einer hohen Unsicherheit möglich. Um dennoch eine Aussagen über diesen Bereich treffen zu können wurde die Messung von  $\tau_w$  erst bei  $I_{\text{radapt}} = 50$  DAC begonnen.

**Startnäherung** Durch das großen Intervall in dem sich  $\tau_w$  bewegen kann, wirkt sich eine konstante Startnäherung besonders negativ auf die Stabilität der Fitroutine aus. Um dies zu verdeutlichen wurden in Tabelle 6 die Mittelwerte über alle Neuronen für eine Messung von  $\tau_w$  mit einer konstanten Startnäherung von  $\tau_{w,S} = 20 \mu\text{s}$  dargestellt. In derselben Tabelle ist der relative Anteil an Messwerten aufgelistet, die unterhalb von 0 oder oberhalb von  $100 \mu\text{s}$  liegen. Die Anzahl dieser abweichenden Messpunkte steigt für kleiner werdende  $\tau_w$  auf bis zu 49 % an. Daraus wird deutlich, dass ein konstantes  $\tau_{w,S}$  nicht optimal geeignet ist um die Adaptionzeitkonstante zu bestimmen. Des Weiteren liegen die Mittelwerte der erfassten Datenpunkte für große  $I_{\text{radapt}}$  weit unter dem im Design vorgegebenen Wert von  $10 \mu\text{s}$ . Man kann also vermuten, dass die tatsächlichen Adaptionzeitkonstanten nicht das geplanten Intervall erreichen und die Startnäherung entsprechend anpassen. Es wurde daher nach einer Funktion gesucht die  $\tau_{w,S}(I_{\text{radapt}})$  sinnvoll beschreiben kann. In Abschnitt 4.2 ist beschrieben, dass die Konduktanz der OTAs, welche zur Implementierung des Adaptionverhaltens genutzt wurden, folgender Proportiona-

lität unterliegen:

$$g_x \sim \sqrt{I_x}.$$

Für eine beliebiges RC-Glied gilt außerdem:

$$\tau_x \sim \frac{1}{g_x} \quad \Rightarrow \quad \tau_x \sim \sqrt{\frac{1}{I_x}}.$$

Die Startnäherung lässt sich demnach durch folgende Funktion beschreiben:

$$\tau_{w,S}(I_{\text{radapt}}) = A \cdot \sqrt{\frac{1}{I_{\text{radapt}}}} + B \quad (26)$$

Wobei A und B Konstanten sind, die durch Randbedingungen bestimmt werden können. Aus Tabelle 6 kann hierzu entnommen werden, dass die kürzeste Adaptionszeitkonstante im Mittel bei  $2,3 \mu\text{s}$  liegt. In Abbildung 16 ist  $\bar{\tau}_w$  sogar  $1,23 \mu\text{s}$ . Aus diesem Grund werden die Randbedingungen folgendermaßen definiert,

$$\tau_{w,S}(50 \text{ DAC}) \stackrel{!}{=} 20 \mu\text{s} \quad \text{und} \quad \tau_{w,S}(950 \text{ DAC}) \stackrel{!}{=} 1,5 \mu\text{s}.$$

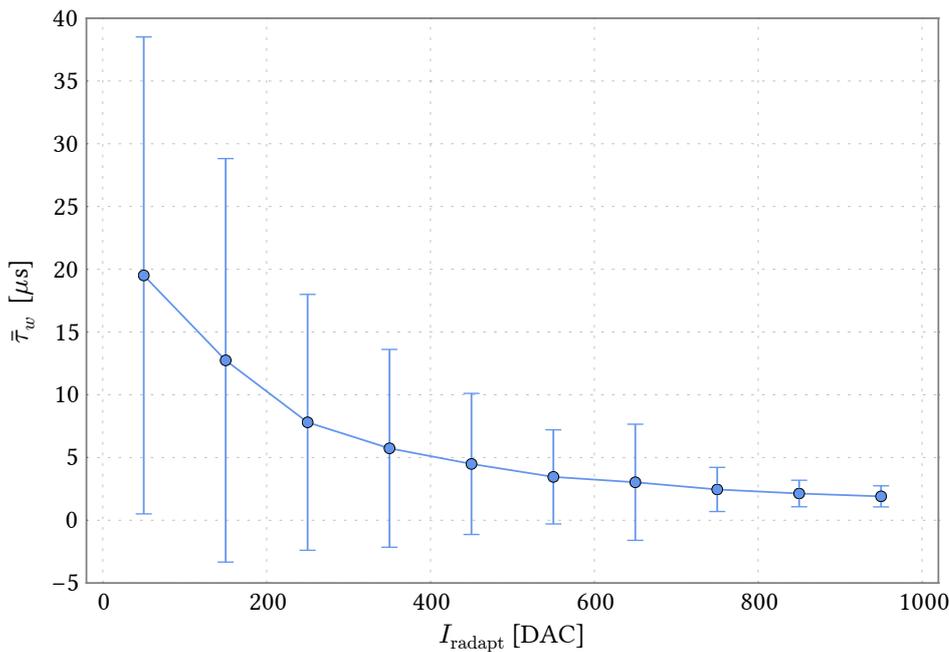


Abbildung 19: Messung von  $\tau_w$  gemittelt über alle Neuronen deren Messwerte zwischen 0 und  $100 \mu\text{s}$  liegen. Die Fehlerbalken beschreiben die Standardabweichung der Einzelmessungen. Es muss jedoch beachtet werden, dass die Verteilungen pro DAC-Wert des Biasstroms nicht symmetrisch sind. In Abbildung 20 ist die Streuung der Messpunkte dargestellt.

Mit diesen Anpassungen der Messroutine wurde nun die Abhängigkeit der Adaptionszeitkonstante vom Biasstrom  $I_{\text{radapt}}$  für alle Neuronen eines HICANN-Chips ermittelt. In

$I_{\text{radapt}} [10^2 \text{ DAC}]$	50	150	250	350	450	550	650	750	850	950
verworfenene $\tau_w$ [%]	38,7	15,6	8,0	3,5	1,2	0,4	0,0	0,0	0,2	0,2

Tabelle 7: Relative Anteile der Messpunkte die bei der Bestimmung von  $\tau_w$ . Werte unterhalb 0 oder oberhalb von 100  $\mu\text{s}$  ergaben.

Abbildung 19 sind die Mittelwerte von  $\tau_w$  dargestellt. Hierbei wurden erneut nur Messpunkte berücksichtigt, die in gewissen Grenzen liegen. Die Messwerte von  $\tau_w$  müssen sich dazu in einem Bereich von 0 bis 100  $\mu\text{s}$  befinden. Der relative Anteil der dadurch verworfenen Datenpunkte ist in Tabelle 5 zusammengefasst. Die Form der Messkurve passt sehr gut zu der theoretischen Vorgabe:  $\tau_w \sim 1/I_{\text{radapt}}$ . Jedoch liegt der gemessene Bereich zwischen 19,5 und 1,9  $\mu\text{s}$  und entspricht damit nicht dem Designintervall von 10 bis 60  $\mu\text{s}$ . Diese Größenordnungen wurden auch mit einer konstanten Startnäherung (siehe Tabelle 6) gefunden, weshalb die eingeführte Näherungsfunktion die Ergebnisse nicht verzerrt haben dürfte. Auffällig ist in dieser Messroutine vor Allem die starke Streuung der Messpunkte. Hierbei muss darauf geachtet werden, dass es sich nicht um Gaußsche oder generell symmetrische Verteilungen handelt. Die Mittelwerte stellen also nicht den Punkt der größten Häufung der Adaptionszeitkonstanten dar.

Um dies genauer zu betrachten, sind in Abbildung 20 exemplarisch die Verteilungen von  $\tau_w$  für drei Werte von  $I_{\text{radapt}}$  visualisiert. Man erkennt, dass die Verteilungen asymmetrisch sind und mit steigenden Biasströmen schmaler werden. Eine mögliche Erklärung hierfür können produktionsbedingte Unterschiede in den OTAs und *floating-gate*-Variationen liefern. Wenn diese Fehlerquellen normalverteilt sind, dann heißt das, dass ein beliebiger Wert  $X$  vom Biasstrom  $I_{\text{radapt}}$  eine symmetrische Streuung über allen Neuronen aufweist. Er besitzt also eine symmetrische Standardabweichung  $\Delta_X$ . Man kann nun die korrespondierenden Werte  $\tau_w(X \pm \Delta_X)$  mithilfe der theoretischen Abhängigkeit der Adaptionszeitkonstante aus Gleichung 26 berechnen. Vergleicht man nun die Differenzen dieser Werte zu  $\tau_w(X)$ , so fällt auf, dass gilt:

$$|\tau_w(X) + \tau_w(X + \Delta_X)| \neq |\tau_w(X) - \tau_w(X - \Delta_X)|$$

Dadurch nimmt der  $1\sigma$ -Bereich um der Mittelwert von  $\tau_w$  eine asymmetrische Form an. Eine solches Verhalten kann in Abbildung 20 betrachtet werden.

Für steigende Biasströme verschmälert sich zudem die Verteilung. Grund hierfür ist, dass die Steigung der Messkurve in Abbildung 19 für große Biasströme abnimmt. Dadurch haben Variationen von  $I_{\text{radapt}}$  einen geringeren Einfluss auf  $\tau_w$ .

Allgemein bestätigen die Messergebnisse der Adaptionszeitkonstante das im Design erwartete Verhalten. Es lässt sich außerdem vermuten, dass die dargestellte Streuung tatsächlichen Variationen des Hardwareparameters  $\tau_w$ . Um dies zu bestätigen, muss zuvor überprüft werden welchen Einfluss die Messunsicherheiten auf die Verteilung der Ergebnisse hat. Dafür sollten Mehrfachmessungen in unterschiedlichen Bereichen von  $I_{\text{radapt}}$  durchgeführt werden. Mithilfe einer genaueren Analyse der Abhängigkeit der Ergebnisse von den Startnäherungen könnten ebenfalls weitere Fehlerquellen ausgeschlossen werden.

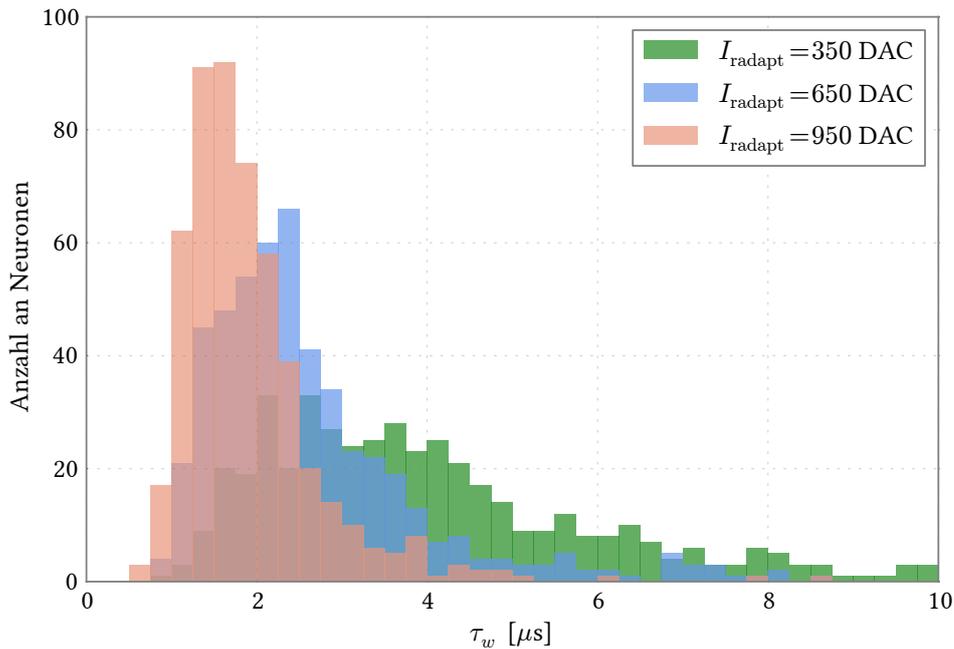


Abbildung 20: Exemplarische Verteilungen von  $\tau_w$  für drei feste Biasstromwerte. Die breiteste Streuung wird durch das kleinste und die schmalste Verteilung durch das größte  $I_{\text{radapt}}$  verursacht. Zur besseren Darstellung werden nur Werte bis zu 10  $\mu\text{s}$  gezeigt.

## 5. Diskussion und Ausblick

Das Ziel dieser Arbeit war es erste Schritte in Richtung einer Implementierung von neuronalen Netzwerken mit Adaptionsverhalten auf dem HICANN-Chip zu machen. Dazu muss eine stabile Kalibrationsebene existieren, welcher zuverlässige Messalgorithmen für die Bestimmung der Adaptionsparameter zur Verfügung stehen. Die Entwicklung und Evaluation solcher Messmethoden wurde in dieser Arbeit durchgeführt.

Im ersten Teil ist dazu eine bereits beschriebene Methode aus *Schwartz (2013)* angepasst und auf ihre Nutzbarkeit für Hardwaremessungen überprüft worden. Es stellte sich dabei heraus, dass die Annahme einer gegen null gehenden Adaptionszeitkonstante auf dem HICANN-Chip nicht gegeben ist.

Daraufhin wurde ein Ansatz, der auf dem Prinzip der Regression basiert und dabei eine endliche Adaptionszeitkonstante berücksichtigt, entworfen. Dieser bietet prinzipiell die Möglichkeit die Parameter  $a$  und  $\tau_w$  simultan zu messen und somit einen zeiteffiziente Kalibration hervorzubringen. Neben dem Fitalgorithmus und den Messprotokollen für die nötigen Membranspannungsverläufe, wurde hierfür ein Simulationsprotokoll entwickelt, das es erlaubt die Genauigkeit und Grenzen der untersuchten Methode zu analysieren. Anhand von dieser Untersuchung konnten Parameterbereiche mit möglichst geringen Unsicherheiten gefunden und Messungen mit den entsprechenden Hardwarekonfigurationen realisiert werden.

Der dynamische Bereich sowie die dargestellten Abhängigkeiten des Adaptionsfaktors  $a$  entsprechen dabei zu hohem Maße den Designvorgaben. Es sollte anhand dieser Mess-

protokolle bereits möglich sein eine Großzahl der Neuronen eines HICANN-Chips an eine Idealkurve zu kalibrieren. Die Messergebnisse der Adaptionzeitkonstante zeigen dagegen eine breitere Streuung. Besonders im Bereich von großen Zeitkonstanten müssen daher Instabilitäten im Fitalgorithmus vermutet werden. Dennoch decken sich auch hier die gewonnen Erkenntnisse über die Adaptionzeitkonstanten weitgehend mit den theoretischen Grundlagen. Interessant ist dabei, dass der dynamische Bereich sehr wahrscheinlich weit unterhalb des im Design beschriebenen Intervalls liegt. Generell ist es durch die dargestellte Messmethode erstmals möglich Referenzwerte für die Adaptionparameter  $a$  und  $\tau_w$  zu gewinnen.

Das Ermittlungsverfahren kann dabei weder in Hinblick auf Zeit noch auf Stabilität als optimiert bezeichnet werden. Bevor an derartigen Verbesserungen gearbeitet wird, muss es jedoch die erste Aufgabe sein die Unsicherheiten und Einschränkungen der Methode genauer zu untersuchen. Besonders die Instabilität bei kleinen  $a$  oder großen  $\tau_w$  schränkt die Anwendbarkeit der Methode stark ein. Hier würde es sich lohnen andere Minimierungsverfahren genauer zu betrachten. Dem derzeit verwendeten Algorithmus können beispielsweise keine Grenzen für die zu optimierenden Parameter gesetzt werden. Eine solche Option könnte jedoch zu einer robusteren Bestimmung des globalen Minimums führen.

Außerdem sollte es möglich sein, anhand der dargestellten Messmethode ein erweitertes Verfahren zur Bestimmung des verbleibenden Adaptionparameters  $b$  zu entwickeln. Wenn  $a$  und  $\tau_w$  bekannt sind, kann die in dieser Arbeit beschriebene Lösungsmethode der AdEx-Gleichungen genutzt werden um eine von  $b$ ,  $a$  und  $\tau_w$  abhängige Funktion  $V(t)$  zu erhalten. Mit einer solchen Funktion könnte man anhand einer Regression im *continuous-spiking*-Regime den letzten fehlenden Adaptionparameter bestimmen.

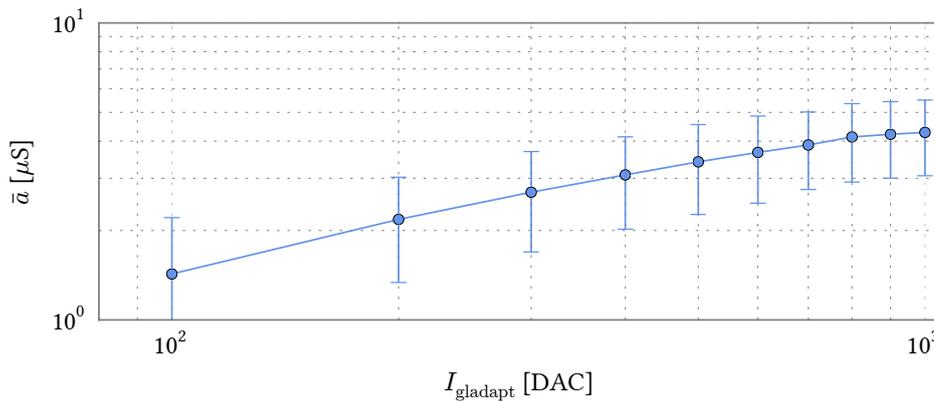
Abschließend lässt sich sagen, dass die ersten Schritte hin zu einem gleichartigen Adaptionverhalten auf dem HICANN-Chip getan sind und die grundlegende Funktionsweise der Hardwareimplementierung bestätigt werden konnte.

## Literatur

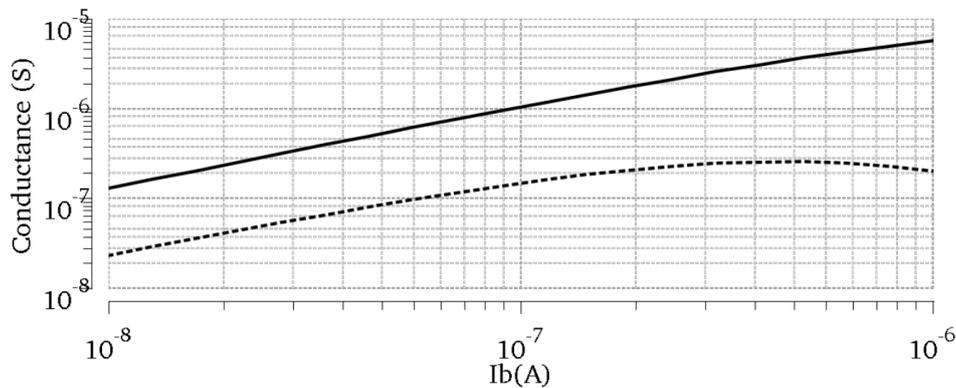
- Binh, T., Demonstrationsexperimente auf neuronmorpher hardware, Bachelor thesis (German), University of Heidelberg, HD-KIP 13-55, 2013.
- BrainScaleS, Research, <http://brainscales.kip.uni-heidelberg.de/public/index.html>, 2012.
- Brette, R., und W. Gerstner, Adaptive exponential integrate-and-fire model as an effective description of neuronal activity, *J. Neurophysiol.*, 94, 3637 – 3642, doi:NA, 2005.
- Denne, M., Testen der software und vermessen des multi-compartment-chips, Bachelor thesis (German), University of Heidelberg, HD-KIP 14-08, 2014.
- EPFL, und IBM, Blue brain project, 2008.
- Fischer, H., und H. Kaul, *Mathematik für Physiker 1*, B. G. Teubner Stuttgart, 1990.
- Gerstner, W., und W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.
- Gewaltig, M.-O., und M. Diesmann, NEST (NEural Simulation Tool), *Scholarpedia*, 2(4), 1430, 2007.
- HBP SP9 partners, *Neuromorphic Platform Specification*, Human Brain Project, 2014.
- Jones, E., T. Oliphant, und P. Peterson, SciPy: Open source scientific tools for Python, 2001.
- Levenberg, K., A method for the solution of certain non-linear problems in least squares, *Quarterly of Applied Mathematics* 2, pp. 164–168, 1944.
- Millner, S., Development of a multi-compartment neuron model emulation, Ph.D. thesis, Ruprecht-Karls University Heidelberg, 2012.
- Schemmel, J., J. Fieres, und K. Meier, Wafer-scale integration of analog neural networks, in *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, 2008.
- Schmidt, D., Automated characterization of a wafer-scale neuromorphic hardware system, 2014.
- Schwartz, M.-O., Reproducing biologically realistic regimes on a highly-accelerated neuromorphic hardware system, Ph.D. thesis, Universität Heidelberg, 2013.

## A. Vergleich des Adaptionfaktors mit Simulationsdaten

In Abbildung 21 ist zu sehen, dass die Messergebnisse von  $a$  einen ähnlichen Verlauf zeigen, wie die Transistorlevel-Simulation der in der Implementierung der Adaption genutzten OTAs. Hierbei ist der dargestellte Intervall der Messpunkte in 21a kleiner als es tatsächlich der Fall ist, da eine Messung bei mit einem Biasstrom von 0 DAC auf einer logarithmischen Achse nicht angezeigt werden kann. Im Bereich großer Werte von  $I_{\text{gladapt}}$  ist der Ansatz eines Plateaus zu erkennen. Eine Erklärung hierfür ist, dass die *floating-gate*-Spannungen, welche den Biasstrom kontrollieren, sich nicht über den gesamten dynamischen Intervall linear verhalten, sondern bei großen DAC-Werten anfangen zu sättigen.



(a) doppelt logarithmische Darstellung der Mittelwerte des Adaptionfaktors. Die Messpunkte entsprechen jenen aus Abbildung 17.



(b) Transistorlevel-Simulation der Konduktanz der im Neuron verwendeten OTAs in Abhängigkeit vom Biasstrom. Die schwarze Linie beschreibt den Mittelwert und die gestrichelte Linie die Standardabweichung einer Iteration über die differentielle Spannung an den Eingängen des OTAs. Entnommen aus *Millner* (2012, S. 39)

Abbildung 21: Die in Abschnitt 4.2 gewonnenen Ergebnisse für  $a$  zeigen eine ähnliche Abhängigkeit vom Biasstrom wie eine Transistorlevel-Simulation des gleichen OTAs beschreibt.

## B. Verwendete Parametersätze

Im Folgendem werden die genauen Parametersätze für die verschiedenen Hardwaremessungen tabellarisch dargestellt. Eine Beschreibung der Parameter kann in *HBP SP9 partners* (2014) nachgeschlagen werden.

Parameter	Wert [DAC]
$E_l$	550
$V_{th}$	500
$V_{reset}$	350
$I_{gl}$	341
$V_{exp}$	1023
$I_{rexp}$	1023
$I_{bexp}$	1023
$I_{gladapt}$	200
$I_{radapt}$	1023

Tabelle 8: Parameter der Messung zu Abbildung 3.

Parameter	Wert [DAC]
$E_l$	550
$V_{th}$	1023
$V_{reset}$	350
$I_{gl}$	341
$V_{exp}$	1023
$I_{rexp}$	1023
$I_{bexp}$	1023
$I_{gladapt}$	0
$I_{radapt}$	0

Tabelle 9: Parameter der Messung zu Abbildung 4.

Simulation		Hardwaremessung	
Parameter	Wert	Parameter	Wert [DAC]
$a$	(0, 0.8, 1.8, 2.7, 3.6) nS	$I_{gladapt}$	0, 80, 250, 500, 1023
$E_l$	-60 mV	$E_l$	550
$V_{th}$	0 mV	$V_{th}$	1023
$V_{reset}$	-60 mV	$V_{reset}$	350
$g_l$	21.6 nS	$g_l$	341
$\Delta_T$	0 mV	$V_{exp}$	1023
		$I_{rexp}$	1023
		$I_{bexp}$	1023
$\tau_w$	100 ns	$I_{radapt}$	1023

Tabelle 10: Parameter zur Messung aus Abschnitt 3.2.2.

Parameter	Wert
$\tau_w$	$10^{-3} \dots 100 \text{ ms}$
$a$	$3 \text{ }\mu\text{S}$
$E_l$	$-60 \text{ mV}$
$V_{\text{th}}$	$0 \text{ mV}$
$V_{\text{reset}}$	$-60 \text{ mV}$
$g_l$	$21.6 \text{ nS}$
$\Delta_T$	$0 \text{ mV}$
$b$	$0 \text{ pA}$

Tabelle 11: Parameter der Messung zu Abbildung 6.

Parameter	Wert [DAC]
$I_{\text{gladapt}}$	$0 \dots 1000$
$E_l$	$480$
$V_{\text{th}}$	$1023$
$I_{\text{gl}}$	$200$
$V_{\text{exp}}$	$1023$
$I_{\text{rexp}}$	$1023$
$I_{\text{bexp}}$	$1023$
$I_{\text{radapt}}$	$1023$

Tabelle 12: Parameter der Messung des Adaptionfaktors  $a$  aus Abschnitt 4.2.

Parameter	Wert
$\tau_w$	$2 \text{ }\mu\text{s}, 10 \text{ }\mu\text{s}$
$\tau_m$	$2,5 \text{ }\mu\text{s}$
$a$	$0,4 \text{ }\mu\text{S}$
$E_l$	$0,76 \text{ V}$
$w_0$	$0,2 \text{ }\mu\text{A}$
$V_0$	$0,09 \text{ V}$
$C_m$	$2,16 \text{ pF}$

Tabelle 13: Parameter zu Berechnung von theoretischen Spannungsverläufen für Validierung der analytischen Lösung aus Abschnitt 3.3.2.

## **Erklärung**

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

Heidelberg, den 10.11.2014