Dynamical System Guided Mapping of Quantitative Neuronal Models Onto Neuromorphic Hardware

Peiran Gao, Ben V. Benjamin, and Kwabena Boahen, Member, IEEE

Abstract—We present an approach to map neuronal models onto neuromorphic hardware using mathematical insights from dynamical system theory. Quantitatively accurate mappings are important for neuromorphic systems to both leverage and extend existing theoretical and numerical cortical modeling results. In the present study, we first calibrate the on-chip bias generators on our custom hardware. Then, taking advantage of the hardware’s high-throughput spike communication, we rapidly estimate key mapping parameters with a set of linear relationships for static inputs derived from dynamical system theory. We apply this mapping procedure to three different chips, and show close matching to the neuronal model and between chips—the Jenson–Shannon divergence was reduced to at least one tenth that of the shuffled control. We confirm that our mapping procedure generalizes to dynamic inputs: Silicon neurons match spike timings of a simulated neuron with a standard deviation of 3.4% of the average inter-spike interval.

Index Terms—Dynamical systems, neural simulation, neuromorphic engineering, quadratic integrate-and-fire model, silicon neuron.

I. QUANTITATIVE NEUROMORPHIC MAPPING

EUROMORPHIC engineering aims to emulate computations carried out in the nervous system by mimicking neurons and their inter-connectivity in VLSI hardware [1]. Having succeeded in morphing visual [2], [3] and auditory [4], [5] systems into mixed-analog-digital circuits, engineers are entering the arena of cortical modeling [6]–[9]. This is an arena in which neuromorphic systems’ parallel operation and low energy consumption give them distinct advantages over software simulation. Reproducing existing theoretical and numerical cortical modeling results using the neuromorphic approach will be facilitated by establishing quantitative links between parameters of neural models and those of their electronic analogs. Furthermore, this will build a foundation for engineers to scale up neuromorphic systems beyond the limit of software simulators.

Initial attempts to map neural models onto neuromorphic chips took a model-less approach [Fig. 1(a)]. Key operating voltages or currents, and occasionally their transients, were measured using on-chip analog-to-digital converters or external instruments. These measurements were compared with the desired behavior, and circuit biases were adjusted using intuition or heuristic algorithms until the matching was acceptable [10], [11]. While such procedures capture the hardware operation in detail, they are time-consuming in both data collection and analysis, and space-consuming in terms of hardware, as special circuits have to be included to expose relevant circuit nodes.

Recent successes in the application of dynamical system theory to computational neuroscience [12] make it now possible to take a model-guided approach to mapping neural models onto neuromorphic hardware. Instead of designing circuits to produce desired behaviors in an ad hoc fashion, designers synthesize current-mode subthreshold CMOS circuits directly from ordinary differential equations that govern the model’s state variables’ evolution, representing these state variables as currents [13]–[15]. This design procedure yields combinations of circuit biases that are mathematically related to neural model parameters through sets of hardware specific mapping parameters. Once these mapping parameters are determined, translating neuronal parameters into circuit parameters is straightforward.

As many neuronal models exhibit complex dynamics, without simple analytical solutions, mapping parameters have so far been extracted through nonlinear optimization, using the model’s numerical simulation as a reference [Fig. 1(b)]. In one approach, the mapping parameters are adjusted iteratively until the circuit’s operating currents converge with the numerical simulation’s state variables. This iterative optimization procedure compensates for hardware discrepancies, starting with an initial set of biases computed from the circuit’s nominal mapping parameters (i.e., obtained from design specifications) [16], [17]. In another approach, the circuit behavior is approximately

---

Manuscript received November 04, 2011; revised December 17, 2011; accepted January 23, 2012. This work was supported by the National Institute of Health (NIH) under Director’s Pioneer Award DPI-OD000965. This paper was recommended by Associate Editor V. Gaudet.

P. Gao and K. Boahen are with the Department of Bioengineering, Stanford University, Stanford, CA 94305 USA (e-mail: pgao@stanford.edu; boahen@stanford.edu).

B. V. Benjamin is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: benvb@stanford.edu).

Digital Object Identifier 10.1109/TCSI.2012.2188956
is the membrane, the quadratic IF neuron obeys to infinity in finite time. When and to 0, we obtain to annihi-
. Solving for the system’s fixed points pushes, approximates the sodium/61
are
In the absence of any fixed points, the neuron spikes repeatedly. (phase portraits in the sub- and supra-threshold regimes, and at bifurcation
fitted by a simpler intermediate model. Then, the intermediate model’s parameters are related to those of the neuronal model through nonlinear optimization of these two models’ simulation results [18]. However, these approaches are time-consuming, and have been restricted to only tens of neurons and their performance has not been demonstrated across multiple chips.
While the complexity of neuron models as a whole seems to merit the nonlinear optimization approach using simulations, analysis of the models in their parts yields simple linear relationships that we exploit here to extract mapping parameters efficiently [Fig. 1(c)]. Mathematical analysis of the neuron model reveals linear scalings of spiking rates and of threshold behaviors that are directly applicable to the analog circuit, since its dynamics match the neuron model’s by design. For example, the silicon neuron’s spiking rate depends nonlinearly on its constant input, which corresponds to ratios of two bias currents scaled by a mapping parameter. A naïve way to find this parameter would be to sweep these currents, record spiking rates, and then perform a nonlinear fit of the analytical solution, if it exists. Alternatively, dynamical system theory predicts that a bifurcation (from resting to spiking) occurs when the constant input is 0.5 (normalized unit). This property of the model constrains the values of the bias currents at bifurcation, which we exploit to obtain the mapping parameter through a linear fit. Similar insights guide our data collection process, and result in a series of simple linear parametric fits for other mapping parameters.
Furthermore, as these linear relationships involve the model’s spiking rate, we can take advantage of the high-bandwidth spike communication in modern neuromorphic systems [19] to achieve high-speed mapping. We demonstrate our method’s efficiency and speed by mapping the quadratic and cubic integrate-and-fire (IF) neuron models (the latter does not have an analytical spiking rate expression) onto tens of thousands of silicon neurons on three different chips from our custom hardware, Neurogrid, proposed in [9].
In Section II, we analyze the quadratic IF neuron model and its implementation in neuromorphic hardware. In Section III, we present the neuron circuit’s bias generator, and argue that its proper calibration is essential for accurate mapping. In Section IV, we apply the high-throughput, spike-based mapping procedure; and we demonstrate its accuracy and performance across different chips in Section V. Finally, in Section VI, we summarize our findings, and propose the procedure’s extension to other neuromorphic circuits such as adaptive, synaptic and ion-channel circuits.

II. SILICON NEURON

The quadratic IF neuron uses a quadratic positive feedback term to model the dynamics of voltage-gated sodium channels [20]; it has become the model of choice for large software simulations [21]. We analyze its bifurcation and spiking rate using dynamical system theory, and show how to match these dynamics in silicon.

A. Quadratic IF Neuron Model

When driven with input \( v_{in} \), the quadratic IF neuron obeys the differential equation

\[
\tau_m \dot{v} = -v + \frac{1}{2} v^2 + v_{in}
\]

(1)

where \( \tau_m \) is the membrane time constant and \( v \) is the membrane potential. The squared term, \( v^2/2 \), approximates the sodium channel’s positive feedback current in biological neurons. The leakage current, \( -v \), has its reversal potential at 0. \( v \) and \( v_{in} \) are normalized by the threshold voltage to be unit-less. This normalization preserves the behavior of the neuron, and helps reduce redundant degrees of freedom in both biophysical models and neuromorphic circuits.

The model is a one-dimensional dynamical system with bifurcation parameter \( \tau_m \). Solving for the system’s fixed points \( v^* \) by setting \( \delta \) to 0, we obtain

\[
v^*_\pm = 1 \pm \sqrt{1 - 2v_{in}}.
\]

As \( v_{in} \) passes 0.5 from below, the system undergoes a saddle node bifurcation [22] in which the two fixed points, \( v^*_\pm \), annihilate each other [Fig. 2(a)]. Below 0.5, the neuron is in the subthreshold regime with two fixed points. The stable fixed point (between 0 and 1) attracts the membrane potential to its resting level. Spikes are absent unless \( v(t) \) is perturbed to above the unstable fixed point (between 1 and 2) [Fig. 2(c)]. Above the bifurcation point of \( \tau_m = 0.5 \), the neuron moves into the supra-threshold regime in which it spikes repeatedly, regardless of \( v(t) \)’s initial conditions (Fig. 2(d)). For each spike, the positive feedback term \( v^2/2 \) pushes \( v(t) \) to infinity in finite time. When \( v(t) \) reaches infinity, the model declares a spike, resets \( v(t) \) to 0, and clamps it there for the refractory period’s duration, \( t_{ref} \).
To calculate the model’s inter-spike interval $T$ in the supra-threshold regime as a function of $v_{th}$, we integrate (1):

$$T - t_{\text{ref}} = \int_{0}^{\infty} dt = \tau_{m} h(v_{th})$$

where

$$h(v_{th}) = \frac{\pi}{2} \arccosh(\sqrt{1 + 2v_{th}}) - \frac{\pi}{2} \arccosh(\sqrt{1 + 2v_{th}}).$$

Inverting $T$, we arrive at the firing rate expression, $f$, for the quadratic IF neuron under supra-threshold drive

$$f(v_{th}) = \frac{1}{\tau_{m} h(v_{th}) + t_{\text{ref}}} \quad \text{for} \quad v_{th} > 0.5.$$  \hspace{1cm} (2)

Qualitatively, $f$ rises sharply from 0 Hz as $v_{th}$ increases beyond the bifurcation value of 0.5 [Fig. 2(b)]. Additionally, we observe two relevant linear scalings under constant $h(v_{th})$: in the absence of $t_{\text{ref}}, 1/f$ scales linearly with $\tau_{m}$, and with fixed $\tau_{m}, 1/f$ scales linearly with $t_{\text{ref}}$.

### B. Circuit Implementation

The quadratic, as well as cubic, IF neuron model is realized in Neurogrid using a subthreshold CMOS circuit (Fig. 3) [11], [15], [23]. Briefly, transistors $M_1$, $M_{10}$ and capacitor $C_{m}$ constitute a current-domain leaky integrator with background input supplied by $M_{10}$; its output $I_{m}$ [analog of the membrane potential variable $v$ in (1)] is scaled quadratically or cubically by $M_{2,3,4,5,6}$ to compute the feedback current, $I_{fb}$. When $I_{fb}$ becomes stronger than the pull-up currents at $V_{th}$, it positively re-inforces $I_{m}$ to quickly pull down $V_{th}$ and drive up $I_{m}$. The circuit emits a digital spike when the strong $I_{m}$ moves REQUEST to $V_{th}$. After being read out, ACK pulses $M_{11}$ to discharge $C_{p}$ to ground. While $M_{12}$ recharges $C_{p}$, $M_{8}$ mimics the refractory period’s effect by clamping $V_{th}$ high and $I_{m}$ close to 0. Circuit dynamics are set by bias currents $I_{\text{leak}}$, $I_{\text{back}}, I_{\text{ref}}$ and $I_{\text{an}}$.

Similar to the design approach described in [15], we analyze the neuron circuit as a dynamical system, but with hardware parameters such as transistor sizing and capacitance explicitly included. In the quadratic regime ($I_{\text{an}} \ll I_{m}$), the circuit’s dynamics match the model’s dynamics [(1)],

$$\frac{d\tau_{m}}{I_{\text{leak}}} = -i_{m} + \frac{1}{2} \frac{P_{m}}{I_{\text{leak}}} + \frac{P_{\text{qua}}}{I_{\text{leak}}}$$  \hspace{1cm} (3)

with equivalent variables

$$\nu \Leftrightarrow i_{m}, \quad \tau_{m} \Leftrightarrow \frac{P_{m}}{I_{\text{leak}}}, \quad v_{m} \Leftrightarrow \frac{P_{\text{qua}}}{I_{\text{leak}}}, \quad t_{\text{ref}} \Leftrightarrow \frac{t_{\text{ref}}}{I_{\text{leak}}}.$$  \hspace{1cm} (4)

Here, $i_{m}$ is $I_{m}$ normalized by $I_{\text{leak}}$ [see (12)]; and $p_{\text{qua}}, p_{\text{qua}}$ are mapping parameters consisting of relative transistor sizing, supply voltage, and capacitance values:

$$p_{\text{qua}} = \frac{C_{m} U_{T}}{h\alpha M_{2}}, \quad p_{\text{beta}} = \frac{2}{\alpha_{M_{0}} + \frac{\alpha_{M_{10}}^{2}}{\alpha_{M_{6}}^{2}}}, \quad p_{\text{pref}} = \frac{C_{V_{dd}}}{\alpha_{M_{12}}^{2}}.$$  \hspace{1cm} (5)

where $\alpha_{M_{L}}$ is transistor $M_{L}$’s $W/L$ ratio divided by that of the biasing PMOS (see Fig. 3) and $U_{T}$ is the thermal voltage. Appendix A details (3)’s derivation; Appendix B shows the same for the cubic IF neuron, which does not have an analytical spiking rate expression.

To map a quadratic IF neuron model with specified model parameters $\tau_{m}, v_{m}$ and $t_{\text{ref}}$, we compute the bias currents

$$I_{\text{leak}} = \frac{P_{m}}{\tau_{m}}, \quad I_{\text{back}} = \sqrt{\frac{v_{m}}{p_{\text{beta}}}}, \quad I_{\text{ref}} = P_{\text{pref}}$$  \hspace{1cm} (6)

which are fed to the biasing transistors (Fig. 3, gray areas) by the on-chip bias generators. Even though the mapping procedure extracts mapping parameters for individual neurons, with all neurons on a Neurogrid chip sharing a single set of bias generators, only the mean values of $p_{\text{qua}}, p_{\text{qua}}$ are used.

1 This part of the circuit is modified from [11] to allow operation at 1.8 V, the original diode-connection around $M_{3}$ was moved to REQUEST.

2 $I_{fb}$ mimics the quadratic (or cubic) positive feedback current exactly only when it is subthreshold.

3 It is impossible to drive $I_{m}$ to infinity in practice. Nevertheless, the circuit closely approximates the model.

4 For the quadratic and cubic regimes, $I_{an}$ is set close to zero or to a large value, respectively, by fixing $V_{th}$ at 0 V or 0.5 V.
III. ON-CHIP BIAS GENERATOR

As our mapping procedure aims to produce quantitatively accurate results on current-biased circuits, the on-chip digital-to-analog bias generators should be calibrated. Additionally, transistor mismatch or fabrication defects should be appropriately compensated for. Before describing the calibration procedure, we first describe the circuit itself.

A. Circuit

Neurogrid’s on-chip bias generator consists of a 12-bit digital-to-analog current converter (modified from [24] to use a RAM instead of a shift register) and an output gain stage (Fig. 4). The digital-to-analog converter (DAC) functions like a PMOS “resistive” divider network [25] (Fig. 4, top half). It receives a proportional-to-absolute-temperature current, \( I_{\text{master}} \), generated by a master bias circuit. Currents at each divider node are then evenly split into two pathways of matching impedances \( (2R) \) realized using identically sized transistors. One of the pathways supplies the current to the next divider while the other is to be selected for output. Ten control bits and their complements (Fig. 4, top gray area) either discard splitter currents through \( I_{\text{res}} \), or merge them into \( I_{\text{div}} \), which feeds the output gain stages. In addition to the ten divider bits, two current mirrors amplify \( I_{\text{master}} \) by two and four times, respectively. Their outputs, when selected, also merge into \( I_{\text{res}} \), giving the DAC 12 bits of resolution. \( I_{\text{div}} \) and \( I_{\text{res}} \) pass into virtual grounds realized with a transconductance amplifier, minimizing the mismatch.

To realize a larger dynamical range, the bias generator provides four gain options, \( d_0, d_1, d_2, \) and \( d_3 \), termed div-gains (Fig. 4, bottom half). To generate the div-gains, mirrored \( I_{\text{div}} \) feeds into another resistive divider network laid out using identical NMOS transistors, with five in parallel for \( R \) or five in series for \( 25R \). The theoretically calculated div-gains \( d_{0,1,2,3} \) are 1, 26, 714, and 35725. A set of digital switches selects one of these possible div-gains for output.

When biasing neuron circuits, the output current of the selected div-gain option is converted to voltage by either a single diode-connected PMOS \( (e.g., I_{\text{ref}}, I_{\text{enl}} \) in Fig. 3) or by two PMOSs in series \( (e.g., I_{\text{enl}}, I_{\text{ack}} \) in Fig. 3) before being mirrored by target transistors. The unselected outputs are simply directed to \( V_{\text{dd}} \) by a digital switch. This selection mechanism, however, introduces asymmetry at the drains of the div-gain output transistors. This asymmetry leads to deviations of \( d_{0,1,2,3}'s \) operational values from their design values.

B. Calibration

Since the DAC design has matured through wide usage in the field [24], [25], the main objective for bias generators’ calibration is to extract div-gain values and characterize the biasing transistors. A set of 106 bias generators are shared by all 65,536 neurons on a Neurogrid chip. However, only one bias generator’s voltage output is exposed to the outside for calibration. Therefore, we assume the others have the same calibration parameters, which is reasonable because bias generators are laid out in close proximity.

We first calibrate the bias generator using the single diode-connected PMOS configuration. We programmed all combinations of DAC values and div-gains, and measured the corresponding output voltages using a high-resistance meter (Keithley 6517A). Since we were essentially measuring the current–voltage relationship of the diode-connected PMOS, we fit the measurements to the Vittoz model [26], which captures transistors’ transition between sub- and supra-threshold operation

\[
i_{\text{th}} = i_0 \ln^2 \left(1 + e^{\frac{(v_T - v_{\text{th}})}{2U_T}}\right)
\]

where \( i_0/\ln^22 \) is the transistor’s current at threshold, \( \kappa \) is the subthreshold slope factor, \( U_T \) is the thermal voltage, and \( v_T \) is the threshold voltage. Additionally, we incorporate \( d_{1,2,3}'s \) values into the model such that a single fit stitches together all four div-gain options. We choose our unit current to be the bias generator’s output when the DAC is set to 1 and div-gain \( d_0 \) is chosen \( (d_0 = 1 \text{ by definition}) \). Conversion to ampere can be done by measuring \( I_{\text{master}} \) (1.1 \mu A, 50 \mu A externally divided by 1/45 mirror internally), but is not necessary for the mapping procedure.

The model fits the data well [Fig. 5(a)] over a current dynamic range of \( 10^8 \) \( (V_{\text{dd}} = 1.8 \text{ V}), v_T \) and \( \kappa \) have fitted values of 0.651 V and 0.701, close to vendor provided numbers. \( d_{1,2,3} \) have values of 31, 910, and 74 015, respectively. The differences between measured and theoretical div-gain values confirms the design imperfection discussed previously, and argues for the necessity of bias generator calibration. When compared across three chips (Table I), \( v_T, d_1, d_2 \) are better matched than \( \kappa, d_3 \). Because neighboring div-gains have overlapping ranges, we set the boundary from one to the other where the first div-gain’s relative fitting error in current surpasses that of the second. Boundaries are at 1.187 V between \( d_{0,1} \), 1.332 V between \( d_{1,2} \) and 1.515 V between \( d_{2,3} \). These voltage boundaries are then converted to current boundaries at 121,026, 4,356, and 0.050
(bias generator units) using the fitted Vitoz model. The fitting error is kept well under 15% for the majority of the bias generator’s operating range.

The calibration results are employed in the following fashion. To set a current bias of 0.15, for example, which falls between the 0.050 and 0.070 of 0.050 at the bifurcation point. At this point, the 0.050 value is then 0.5 at the bifurcation point. At this point, 0.1 to 0.01, decrementing by a ratio of 0.95. Bias combinations at bifurcation—defined as the boundary between spiking and no spiking (in a one-second window)—were recorded for each neuron, in the absence of refractory periods [Fig. 6(a)].

We estimated the neurons’ individual $p_{\text{fore}}$ by linearly fitting values of 0.5$R_{\text{back}}^2$ against those of $I_{\text{back}}^2$ at the bifurcation points for each neuron, in the absence of refractory periods [Fig. 6(a)]. Fitted $p_{\text{fore}}$ for the entire chip’s neurons have a mean of 4.413, with standard deviation 0.992 [Fig. 6(b)]. The distribution has a slightly heavy tail, with 95.4% of the fitted values within two standard deviations of the mean. The relatively broad distribution has a coefficient of variation (CV) of 0.225. We checked the intercepts of the linear fittings, and found them tightly distributed around zero with mean $-0.470 \times 10^{-3}$ and standard deviation $0.410 \times 10^{-5}$, confirming the linear scaling of $I_{\text{back}}^2$ against $R_{\text{back}}^2$ at bifurcation. To verify that $p_{\text{fore}}$’s mean value is truly descriptive for the entire chip, we confirmed that no spatial gradients were present; the spatial distribution is white-noise like. [Fig. 6(c)].
B. Membrane Time Constant

Next, we estimate the mapping parameter $p_{\text{tm}}$ by utilizing the scaling of the circuit’s firing rate with its membrane time constant in the absence of refractory period [(2), (4)]

$$\frac{1}{f} = \frac{h(v_{\text{in}})}{p_{\text{tm}}} = \frac{h(v_{\text{in}})}{I_{\text{leak}}}, \quad v_{\text{in}} > 0.5, \quad t_{\text{ref}} = 0$$

Although $p_{\text{tm}}$’s average value is employed for mapping neuronal models, using individually estimated $p_{\text{tm}}$s for each neuron to compute $v_{\text{in}}$ when given $I_{\text{leak}}$, avoids carrying over errors caused by averaging in one parameter into another. Linearly fitting $1/f$ against $h(v_{\text{in}})/I_{\text{leak}}$ then yields $p_{\text{tm}}$ as slopes [Fig. 7(a)]. Specifically, we stepped $v_{\text{in}}$ between 1 and 11 and $I_{\text{leak}}$ between 0.02 and 0.2, and measured spiking rates in one-second time windows.

Fitted $p_{\text{tm}}$s have a tighter distribution than that of $p_{\text{th}}$, with $CV = 0.072$. The distribution has a mean of $1.346 \times 10^{-3}$ with standard deviation $0.097 \times 10^{-3}$ [Fig. 7(b)]; 96.7% of the neurons are within two standard deviations of the mean. The spatial distribution of each neuron’s $p_{\text{tm}}$’s relative error (with respect to the average) exhibits less noise than that of $p_{\text{th}}$, and is also without a spatial gradient [Fig. 7(c)]. In this case, intercepts are significantly different from zero, with mean $-1.325 \times 10^{-3}$ and standard deviation $0.387 \times 10^{-3}$. These nonzero intercepts are unlikely due to insufficient data or inaccurate measurements of rates, since each linear fit uses approximately 170 data points all above 20 Hz. Exact causes are difficult to pinpoint. One possible explanation may be that $I_{\text{in}}$ does not reach infinity during each spike or zero after each reset. For example, when $v$ integrates only from 0.1 to 100 during a spike, measured $1/f$, when compared against theory, would be 1.6% and 3.0% lower for small and large $v_{\text{in}}$ values, respectively. This may have resulted in a systematic lowering of $1/f$ that gave rise to the negative intercepts. This effect is only salient when $I_{\text{leak}}$ is large, or when $\tau_{\text{m}}$ is sub-millisecond, and is negligible for $\tau_{\text{m}}$ in the physiological range (>1 ms).

C. Refractory Period

Having estimated individual neuron’s $p_{\text{th}}$s and $p_{\text{tm}}$s for the chip, we can calculate their spiking rates when the refractory period is zero: $f_0 = 1/(\tau_{\text{th}} h(v_{\text{in}}))$. $f_0$ is related to $f$, the spiking rate with nonzero refractory period, by [(2), (4)]

$$\frac{1}{f} - \frac{1}{f_0} = \frac{p_{\text{ref}}}{T_{\text{ref}}}$$

An individual neuron’s $p_{\text{ref}}$ is then the slope of a linear $1/f = 1/f_0$ versus $1/T_{\text{ref}}$ fit [Fig. 8(a)]. Specifically, we stepped $f_0$ from 20 to 220 Hz, by choosing $v_{\text{in}}$ between 5 and 20, and $\tau_{\text{m}}$ between 5 and 40 ms, swept $T_{\text{ref}}$ from 1.25 to 50, and measured spiking rates in 1-s time windows.
Fitted \( p_{ref} \) have the tightest distribution among the three mapping parameters, with mean \( 26.565 \times 10^{-3} \), standard deviation \( 1.478 \times 10^{-3} \) and \( CV = 0.055 \); 95.4% of the fitted values are within two standard deviations of the mean [Fig. 8(b)]. As a result of the small CV, the spatial distribution of each neuron’s \( p_{ref} \)’s relative error (with respect to the average) is also the least noisy among the mapping parameters, with no gradient [Fig. 8(c)]. To a lesser extent than \( p_{ref} \), the fits’ intercepts also have a slight negative bias: \(-2.298 \times 10^{-3} \pm 1.706 \times 10^{-3} \). This is again unlikely due to insufficient data or erroneous rate measurements, as approximately 200 data points were collected for each neuron. It may be attributed to the same cause of incomplete \( I_{mem} \) reset discussed previously.

D. Generalization to Dynamic Input

Since the above mapping parameters cover all degrees of freedom between the model and the circuit, the mapped neurons should accurately generalize to behaviors other than tonic spiking, which we confirmed by applying dynamic inputs. We mapped a simulated neuron (\( \tau_m = 10 \) ms, \( t_{ref} = 5 \) ms) to a silicon neuron using its own mapping parameters, and compared the time courses of their responses (Fig. 9). The model’s dynamics \([11]\) was simulated in Python with spikes declared at \( v = 100 \) and integration step 100 μs. Unlike the static input used in parameter estimation, both neurons received a dynamic input \( \tau_{syn}(t) \) generated by drawing from a uniform distribution between 0.5 and 0.9 every 60 ms (Fig. 9 top). The silicon neuron’s response \( (I_{mem}) \) is measured using a 10-bit asynchronous on-chip analog-to-digital converter (ADC), sampling on average at 1.8 kHz. As the ADC’s gain was not calibrated, we scale its output to match the simulated neuron’s response. The mapped neuron’s response is noisy, but closely matches the simulated neuron’s (Fig. 9 bottom); the average spiking rate was 10 Hz. The standard deviation in spike-time difference between the simulated and ten mapped neurons was 3.4 ms—3.4% of the average inter-spike interval—for 30 spikes per neuron (average in a three-second window). This faithful reproduction confirms that our static-input based mapping procedure generalizes to dynamic input.

V. HIGH-THROUGHPUT MAPPING ACROSS CHIPS

To apply the procedure to multiple chips, we further improve the throughput of the parameter mapping procedure by subsampling neurons. We use subsampling to rapidly map the quadratic IF neuron onto three chips (labeled as in Table I), and show highly consistent mapped neuron behaviors.

A. Neuron Subsampling

Despite the high rate of spike communication using the address event representation, we can only estimate the mapping parameters for a fraction of the 65,536 neurons at a time (the remaining neurons are disabled). While individual chips in the Neurogrid system communicate up to 90 M spikes per second with each other, communication to desktop computers through USB ports only supports around 3.2 M spikes per second.\(^5\) For maximal stability during data collection, we utilize a bandwidth

\(^5\)Each spike packet consists of 5 words of 16 bits. Maximum USB transfer is rated at 32 MB per second. More spikes may be communicated using the burst-mode address event representation [27].
of 1-M spikes per second. During our mapping procedure, neurons emit up to 250 spikes per second, allowing 4096 neurons to be simultaneously recorded. Therefore, we sought to ascertain if equally accurate mapping results can be obtained by subsampling randomly and uniformly.

We quantify subsampling’s impact on mapping parameter estimation accuracy by computing the relative standard error in mean for all three parameters for different sample sizes (Fig. 10). The relative standard error in mean roughly scales as the inverse of the square root of sampling sizes (Fig. 10 inset), confirming the law of large numbers.

of 0.007 or less, indicating closely matched behaviors across chips (Fig. 11 insets). Between Chip B and C for \( \tau_m = 20 \text{ ms} \) and \( t_{\text{ref}} = 5 \text{ ms} \), the measure has a value of 0.017. On the other hand, for the shuffled control, we mapped neurons onto Chip A using mapping parameters for both Chip A and B since they are more similar with each other than with Chip C. The Jenson–Shannon divergence measure between the two cases yields a value of 0.160 (Fig. 12), an order of magnitude greater than the worst case scenario when using the correct mapping parameters. This large improvement is explained by the large variation in some mapping parameters across chips (see Table II).

Furthermore, deviations between estimated hardware mapping parameters and design values question the assumption that current scales linearly with sizing (from biasing to circuit transistors). While parameters \( p_{\text{tau,ref}} \) have similar values across chips, they are around one sixth that of the design value. Whereas \( p_{\text{qua}} \) is within a factor of two of its design value, its large cross-chip variation. We note that \( p_{\text{qua}} \)’s expression has transistor sizing ratios \( \sigma_{\text{MgS}} \) with identical powers in both the denominator and numerator, while those of \( p_{\text{tau,ref}} \) differ by one \([5]\). The net power difference suggests that the systematic factor of six in \( p_{\text{tau,ref}} \) and \( p_{\text{ref}} \) arises from the failed assumption that current ratios between biasing and circuit transistors are the same as their sizing ratios. Interestingly, the relative ratios between \( p_{\text{tau}} \) and \( p_{\text{ref}} \) are similar for both hardware and design values.

Last but not least, spiking rate distributions under different model parameters lend insight into the differential contribution to behavior mismatch by different parts of the circuit. For the same \( \tau_m \) and \( t_{\text{ref}} \), spiking rate variations across neurons show no dependence on \( \tau_m \), differing by magnitude \([6]\). Linear fits of spiking rates’ standard deviations against \( \tau_m (\tau_m \geq 0.7 \text{ only}) \) only failed with coefficient of determination \( R^2 \) values of 0.003, 0.041, and 0.034 for the three parameter combinations. Mappings using refractory periods of 5-ms show decreased spiking rate variations when compared to those with refractory periods of 0 ms, implying that transistors that implement the refractory period alleviate cross-neuron mismatch (intrachip).

VI. DISCUSSION

We presented a dynamical-system-guided procedure to map quantitatively accurate neuronal models onto neuromorphic chips. We applied mathematical characterization rooted in dynamical systems to neuromorphic circuits designed to match model dynamics, and formulated the mapping procedure as three simple linear least-square fits. The mapping procedure achieved high throughput by taking advantages of high-bandwidth spike communication and subsampling the hardware...
and closely follow the neuronal model’s predicted behaviors: combinations are each mapped to three different chips. (a) percentile with whiskers. plots) of all 65 536 neurons’ spiking rates at different values of Chip A’s original mapping parameters and those of Chip B. Distributions (box plots) for all 65 536 neurons’ spiking rates at different values of \( \tau_m \) closely follow the neuronal model’s predicted behaviors (solid lines) across different chips (different box shades). The pair-wise Jenson–Shannon divergence measure quantifies the interchip differences (gray insets). The box plots represent distributions’ medians with central bars; their central 50 percentile with boxes (shades distinguish different chips); and their 5th and 95th percentile with whiskers.

Fig. 11. Quadratic IF neuron model mapping across chips. Instances of the quadratic IF neuron model with three different membrane time constant \( \tau_m \) and refractory period \( t_{ref} \) combinations are each mapped to three different chips. (a) \( \tau_m = 10 \text{ ms}, t_{ref} = 0 \text{ ms} \). (b) \( \tau_m = 10 \text{ ms}, t_{ref} = 5 \text{ ms} \). (c) \( \tau_m = 20 \text{ ms}, t_{ref} = 5 \text{ ms} \). The distributions (box plots) for all 65 536 neurons’ spiking rates at different values of \( \tau_m \) are compared against the theoretical behavior. The incompatible mapping parameter set results in mapped neuron behaviors significantly different from the theoretical prediction (solid line). The Jenson–Shannon divergence measure between the two distributions (gray inset) quantifies the difference.

neuron populations. It accurately generalized to neuron behaviors other than tonic spiking, and demonstrated cross-chip consistency on three Neurogrid chips. The procedure also quantifies heterogeneity in our large-scale system, a hardware feature deserving further investigation. When applying our mapping procedure to architectures that do not share biases among neurons, one can achieve exact matching by setting each neuron’s biases using its own mapping parameters (Figs. 6–8). However, subsampling would be no longer viable, therefore, higher-throughput spike communications would be required to map every neuron concurrently. We also caution against the excessive overhead of calibrating each neuron’s bias generator set, and suggest that a balancing act between time and accuracy may be required.

Furthermore, the mapping procedure can be readily generalized to additional components of the neuron circuit such as spike frequency adaptation. In a published design of the adaptation circuit [15], each spike increments the leakage conductance by a fixed amount, which decays to zero with a finite time constant in the absence of spikes. At steady state, the adaptive leakage conductance is proportional to the spiking rate. Hence, a linear fit yields the proportionality constant, which is the product of the fixed increment and the time constant. Finally, these two parameters can be resolved by analysis of the first few inter-spike intervals, yielding the corresponding mapping parameters.

To pave the path for future large-scale cortical modeling on neuromorphic systems, our mapping procedure will have to extend to more complex circuits. Mapping of only neuronal models is necessary but not sufficient for the construction of quantitatively accurate neural networks. As synapses constitute inter-neuronal connections in networks, their mapping is a priority. Fortunately, the mathematical relationship between neurons’ input spike train statistics and their output spiking behaviors have been solved theoretically for many IF model types [29], [30], including the quadratic IF model [31], linking synaptic parameters to input statistics to output spiking rates, potentially enabling similar spiking-based mapping procedures.

Recent silicon neurons have also been designed to include Hodgkin–Huxley type voltage-gated channel dynamics [32]–[35]. Such channels exhibit complex temporal dynamics when interacting with the neuron’s membrane potential, and participate in spiking. Bifurcations, as well as spiking rates, are related to channel parameters in more complicated ways. While simple linear relations in analyzed models may not be available in more complex dynamics, thorough analysis may still yield mathematical relations that permit for efficient and scalable mapping approaches.

APPENDIX A

QUADRATIC IF NEURON CIRCUIT ANALYSIS

For conciseness, we define the sizing ratios between each transistor \( M_F \) and its biasing transistor as \( \alpha_{M_F} \) (e.g., \( I_{M_F}/I_{M_{bias}} = I_{M_{bias}} \)). Actual sizes are given in Table III. The chip was fabricated in 180-nm CMOS. Thick-oxide transistors were used, except for capacitors, which were thin-oxide transistors.

We first ignore the refractory circuit, and apply Kirchoff’s Current Law to node \( V_m \):

\[
C_m V_m = I_{M_F} - I_{B_F} - I_{M_{bias}}.
\]

(8)
Using the $\alpha_{M_0}$ parameters and the translinear principle [36], we relate $I_{M_7 M_{10}}$ with their bias currents

$$\frac{I_{M_7}}{\alpha_{M_7}} = I_{\text{sink}}$$

$$\frac{I_{M_7} I_m}{\alpha_{M_7} \alpha_{M_4}} = I_{\text{back}}.$$  

Next, to derive $I_{\text{ref}}$, we set $\alpha_{M_5} = 1$, as $M_5$ is actually voltage-biased. Additionally, in the hardware design, $\alpha_{M_5} = \alpha_{M_6}$ and $\alpha_{M_4} = \alpha_{M_7}$. As $M_4$ operates in the ohmic region, we express its current as the difference between a source component and a drain component

$$I_{M_4} = I_{0_{M_4} e^{-\frac{\alpha_{M_5}}{U_T}}} - I_{0_{M_4} e^{-\frac{\alpha_{M_5}}{U_T}}} = I_m - \frac{\alpha_{M_4}}{\alpha_{M_5}} I_{\text{b}}$$

where $U_T$ is the thermal voltage, and $I_{0_{M_4}}$ is $M_4$’s off current. Balancing currents at node $V_\text{d}$, we have

$$I_m = \frac{\alpha_{M_4}}{\alpha_{M_5}} I_{\text{f}} + I_{\text{b}} + I_{M_5}.$$  

Combining the above expression with the translinear relation

$$\frac{I_{M_5} I_m}{\alpha_{M_5} \alpha_{M_4}} = \frac{I_{\text{an}} I_{\text{b}}}{\alpha_{M_5} \alpha_{M_6}}$$

we obtain

$$I_{\text{f}} = \frac{I_m}{1 + \frac{\alpha_{M_5}}{\alpha_{M_6}} (1 + \frac{I_{m}}{I_{\text{f}}})}.$$  

For the quadratic neuron, we substitute (9)-(11) into (8), set $I_{\text{an}} = 0$, and use the fact $V_{\text{ref}} = -\frac{I_{m}}{I_{\text{f}}} U_T / k / I_{\text{f}}$

$$\frac{C_m U_T}{\kappa \alpha_{M_7} I_{\text{sink}}} \frac{I_m}{I_{\text{f}}} = -I_{m} + \left( \frac{I_{m}}{1 + \frac{\alpha_{M_5}}{\alpha_{M_6}}} \right) + \frac{\alpha_{M_6} \alpha_{M_7}}{\alpha_{M_7} I_{\text{sink}}} I_{\text{sink}},$$

We employ one additional substitution to turn this expression into its unit-less form [11]:

$$i_m = \frac{I_m}{2 \left( 1 + \frac{\alpha_{M_5}}{\alpha_{M_6}} \right) \alpha_{M_7} I_{\text{sink}}}$$

and we obtain

$$\frac{p_{\text{ref}}}{I_{\text{sink}}} i_m = -i_m + \frac{1}{2} i_m^2 + p_{\text{quadratic}} I_{\text{sink}}^2.$$  

with parameters

$$p_{\text{q}} = \frac{C_m U_T}{\kappa \alpha M_7} \quad \text{and} \quad p_{\text{ref}} = \frac{\alpha_{M_4} \alpha_{M_10}}{2 \left( 1 + \frac{\alpha_{M_5}}{\alpha_{M_6}} \right) \alpha_{M_7}^2}.$$  

When an emitted spike is acknowledged, $M_{11}$ quickly sets $V_\text{r}$ to ground. The duration of the refractory period is then the time taken by $M_{12}$ to recharge $C_T$. Hence, we have a simple expression

$$t_{\text{ref}} = \frac{p_{\text{ref}}}{I_{\text{ref}}} \quad \text{where} \quad p_{\text{ref}} = \frac{C_T V_{\text{del}}}{\alpha_{M_12}}.$$  

**APPENDIX B**

**MAPPING OF THE CUBIC IF NEURON**

The cubic IF neuron follows the dynamics

$$m \frac{\tau_m \dot{\nu}}{\tau_m} = -v + \frac{1}{3} v^3 + v_{\text{in}},$$

Similar to the quadratic IF neuron, a saddle-node bifurcation takes place when $v_{\text{in}}$ exceeds 2/3 (from rest to spiking). The spiking rate in the absence of refractory period is not analytical, and has the numerical expression

$$f = \frac{1 - r^2}{\tau_m \ln(-r)}$$

where $r$ is the root of the polynomial

$$-v + \frac{1}{3} v^3 + v_{\text{in}} = 0$$

Note that there is only one negative value for $r$ given that $v_{\text{in}} > 2/3$ (supra-threshold).

The neuron circuit mirrors the cubic IF model when $\alpha_{M_4} I_{\text{an}} \gg \alpha_{M_6} I_m$. Using this relation and (11), $I_{\text{b}}$ becomes

$$I_{\text{b}} = \frac{\alpha_{M_6} I_{\text{an}}}{\alpha_{M_4}}$$

Note that as $I_m$ increases during a spike, $I_{\text{b}}$’s expression starts to deviate from this approximation. Defining a different substitution for $I_m$

$$i_m' = \sqrt{\frac{3 \alpha_{M_4}}{\alpha_{M_4} \alpha_{M_7} \alpha_{M_10} I_{\text{sink}}}} I_m$$

we arrive at the cubic form

$$\frac{C_m U_T}{\kappa \alpha_{M_7} I_{\text{sink}}} \frac{I_m'}{I_{\text{sink}}} = -I_{m'} + \frac{1}{3} i_{m'}^2 + p_{\text{cubic}} I_{\text{sink}}^2$$

with

$$p_{\text{cubic}} = \frac{\alpha_{M_4} \alpha_{M_10}}{\alpha_{M_7}} \sqrt{\frac{3 \alpha_{M_4}}{\alpha_{M_4} \alpha_{M_7} I_{\text{an}}}}.$$  

Note that, unlike $p_{\text{q}}$, $p_{\text{cubic}}$ is not unit-less.

Using the estimated value of $p_{\text{q}}$ from the quadratic IF neuron mapping, we only have to estimate $p_{\text{cubic}}$ for the cubic

6Operationally, we set a large constant value for $I_{\text{an}}$ by fixing $V_{\text{an}}$ at 0.5 V.
model. Similarly to $p_{\text{th}}$, we take advantage of the circuit’s behavior at bifurcation.

$$v_{\text{in}} \Leftrightarrow p_{\text{th}} \frac{\tau_{\text{dec}}^2}{\tau_{\text{leak}}^2} = \frac{2}{3}$$

and linearly fit $(2/3)^{\frac{1}{2}}\tau_{\text{leak}}$ against $\frac{\tau_{\text{dec}}^2}{\tau_{\text{leak}}^2}$ to obtain $p_{\text{th}}$. $p_{\text{th}}$ has values 0.357, 0.406, and 1.403 for chips A, B, and C. We measured spiking rates as a function of $v_{\text{in}}$ after mapping cubic neurons to the three chips with $\tau_{\text{in}} = 10$ ms and $t_{\text{ref}} = 5$ ms (Fig. 13). Similar to the quadratic IF neuron, data from the three chips again show excellent agreement with each other and thresholding behavior at $v_{\text{in}} = 2/3$. However, measurements are lower than theoretical calculations for low spiking rates. In this regime, $I_{\text{in}}$ increases slowly before each spike, resulting in the circuit violating the assumption $\alpha_{\text{M}} I_{\text{in}} \gg \alpha_{\text{M}} I_{\text{in}}$ for a significant portion of the inter-spike interval. This violation reduces $I_{\text{in}}$ from a cubic to a quadratic function of $I_{\text{in}}$.

ACKNOWLEDGMENT

The authors would like to thank E. McQuinn for designing and programming the graphical user interface used for collecting spiking rates from mapped neurons. The authors would also like to thank J. Arthur, P. Merolla, and D. Neil for supporting the Neurogrid hardware.

REFERENCES

[31] N. Brunel and P. E. Latham, “Firing rate of the noisy quadratic inte-

brane current and its application to conduction and excitation in nerve,”


[34] K. M. Hynna and K. Boahen, “Thermodynamically-equivalent silicon
2007.

of analog operators based on the Hodgkin-Huxley formalism for the

Jenkins, and K. Strohbehn, “Current-mode subthreshold MOS circuits

**Peiran Gao** received the B.A. degree in neurobiology and physics with a minor in electrical engineering and computer science from University of California, Berkeley, and the M.S. degree in bioengineering from Stanford University, Stanford, CA, in 2009 and 2011, respectively.

He is currently a Graduate Student in bioengineering at Stanford University, where, as the Teaching Assistant for the BIOE 332 Large-Scale Neural Modeling course in Spring 2011 Quarter, he developed the dynamical system guided mapping procedure detailed in this study. His research focuses on the theoretical development of a spike-based computational framework.

**Ben V. Benjamin** received the B.Tech. degree in electronics and communication engineering from Mahatma Gandhi University, Kerala, India, in 2005 and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2010. He is currently pursuing the Ph.D. degree at Stanford University.

He led the design and implementation of software drivers used to program and communicate with Neurogrid for the mapping procedure. Thereafter, he worked for three years as a Design Engineer in the VSVU group at Wipro Technologies, India, where he earned the Prodigy award for the best incoming employee of the year. While there, he also received a patent for his work on standard cell characterization.

**Kwabena Boahen** (M’89) received the B.S. and M.S.E. degrees in electrical and computer engineering from the Johns Hopkins University, Baltimore MD, in 1989 and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, in 1997.

He is an Associate Professor in the Bioengineering Department at Stanford University. He is a Bioengineer who is using silicon integrated circuits to emulate the way neurons compute, linking the seemingly disparate fields of electronics and computer science with neurobiology and medicine. His contributions to the field of neuromorphic engineering include a silicon retina that could be used to give the blind sight and a self-organizing chip that emulates the way the developing brain wires itself up. His scholarship is widely recognized, with over seventy publications to his name, including a cover story in the May 2005 issue of Scientific American.

From 1997 to 2005, he was on the faculty of the University of Pennsylvania, Philadelphia PA, where he held the first Skirkanich Term Junior Chair.

Prof. Boahen has received several distinguished honors, including a Fellowship from the Packard Foundation (1999), a CAREER award from the National Science Foundation (2001), a Young Investigator Award from the Office of Naval Research (2002), and the National Institutes of Health Directors Pioneer Award (2006). He is a member of Tau Beta Kappa. He held a Sloan Fellowship for Theoretical Neurobiology at the California Institute of Technology.