Extending the Neural Engineering Framework for Nonideal Silicon Synapses

Aaron R. Voelker*, Ben V. Benjamin†, Terrence C. Stewart*, Kwabena Boahen† and Chris Eliasmith*
{arvoelke, tcstewar, celiasmith}@uwaterloo.ca {benvb, boahen}@stanford.edu
*Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada.
†Bioengineering and Electrical Engineering, Stanford University, Stanford, CA, U.S.A.

***Abstract*—The Neural Engineering Framework (NEF) is a theory for mapping computations onto biologically plausible networks of spiking neurons. This theory has been applied to a number of neuromorphic chips. However, within both silicon and real biological systems, synapses exhibit higher-order dynamics and heterogeneity. To date, the NEF has not explicitly addressed how to account for either feature. Here, we analytically extend the NEF to directly harness the dynamics provided by heterogeneous mixed-analog-digital synapses. This theory is successfully validated by simulating two fundamental dynamical systems in Nengo using circuit models validated in SPICE. Thus, our work reveals the potential to engineer robust neuromorphic systems with well-defined high-level behaviour that harness the low-level heterogeneous properties of their physical primitives with millisecond resolution.**

I. The Neural Engineering Framework

The field of neuromorphic engineering is concerned with building specialized hardware to emulate the functioning of the nervous system [1]. The Neural Engineering Framework (NEF; [2]) compliments this goal with a theory for "compiling" dynamical systems onto spiking neural networks, and has been used to develop the largest functioning model of the human brain, capable of performing various perceptual, cognitive, and motor tasks [3]. This theory allows one to map an algorithm, expressed in software [4], onto some neural substrate realized in silicon [5]. The NEF has been applied to neuromorphic chips including Neurogrid [5], [6] and a VLSI prototype from ETH Zurich [7].

However, the NEF assumes that the *postsynaptic current* (PSC) induced by a presynaptic spike is modelled by a first-order lowpass filter (LPF). That is, by convolving an impulse representing the incoming spike with an exponentially decaying impulse-response. Furthermore, the exponential time-constant is assumed to be the same for all synapses within the same population. In silicon, synapses are neither first-order nor homogeneous and spikes are not represented by impulses.[1] Synapse circuits have parasitic elements that result in higher-order dynamics, transistor mismatch introduces variability from circuit to circuit, and spikes are represented by pulses with finite width and height. Previously, these features restricted the overall accuracy of the NEF within neuromorphic hardware (e.g., in [5], [6], [7]).

The silicon synapses that we study here are mixed-analog-digital designs that implement a pulse-extender [9] and a first-order LPF [10], modelled as a second-order LPF to account for parasitic capacitances. We also account for the variability (i.e., heterogeneity) introduced by transistor mismatch in the extended pulse's width and height, and in the LPF's two time-constants. In §II, we demonstrate how to extend the NEF to directly harness these features for system-level computation. This extension is tested by software simulation in §IV using the circuit models described in §III.

II. Extending the Neural Engineering Framework

The NEF consists of three principles for describing neural computation: representation, transformation, and dynamics [2]. This framework enables the mapping of dynamical systems onto recurrently connected networks of spiking neurons. We begin by providing a self-contained overview of these three principles using an ideal first-order LPF. We then extend these principles to the heterogeneous pulse-extended second-order LPF, and show how this maps onto a target neuromorphic architecture.

A. Principle 1 – Representation

The first NEF principle states that a vector $\mathbf{x}(t) \in \mathbb{R}^k$ may be *encoded* into the spike-trains δ_i of n neurons with rates:

$$r_i(\mathbf{x}) = G_i\left[\alpha_i \mathbf{e}_i \cdot \mathbf{x}(t) + \beta_i\right], \quad i = 1 \dots n, \tag{1}$$

where G_i is a neuron model whose input current to the soma is the linear encoding $\alpha_i \mathbf{e}_i \cdot \mathbf{x}(t) + \beta_i$ with gain $\alpha_i > 0$, unit-length encoding vector $\mathbf{e}_i$ (row-vectors of $\mathbf{E} \in \mathbb{R}^{n \times k}$), and bias current β_i. The state $\mathbf{x}(t)$ is typically *decoded* from spike-trains (see Principle 2) by convolving them with a first-order LPF that models the PSC triggered by spikes arriving at the synaptic cleft. We denote this filter as $h(t)$ in the time-domain and as $H(s)$ in the Laplace domain:

$$h(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}} \quad \Longleftrightarrow \quad H(s) = \frac{1}{\tau s + 1}. \tag{2}$$

Traditionally the same time-constant is used for all synapses projecting to a given population.[2]

B. Principle 2 – Transformation

The second principle is concerned with decoding some desired vector function $\mathbf{f} : S \to \mathbb{R}^k$ of the represented vector. Here, S is the domain of the vector $\mathbf{x}(t)$ represented via Principle 1—typically the unit k-cube or the unit k-ball. Let $r_i(\mathbf{x})$ denote the expected firing-rate of the i^{th} neuron in

[1] These statements also hold for real biological systems [8].

[2] See [11] for a recent exception.

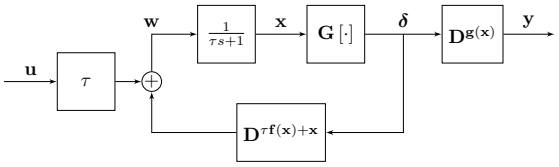

Fig. 1. Standard Principle 3 (see (6)) mapped onto an ideal architecture to implement a general nonlinear dynamical system (see (5)). The state-vector $\mathbf{x}$ is encoded in a population of neurons via Principle 1. The required signal $\mathbf{w}$ is approximated by $\tau\mathbf{u}$ plus the recurrent decoders for $\tau\mathbf{f}(\mathbf{x})+\mathbf{x}$ applied to $\boldsymbol{\delta}$, such that the first-order LPF correctly outputs $\mathbf{x}$. The output vector $\mathbf{y}$ is approximated using the decoders $\mathbf{D}^{\mathbf{g}(\mathbf{x})}$.

response to a constant input $\mathbf{x}$ encoded via (1). To account for noise from spiking and extrinsic sources of variability, we introduce the noise term $\eta \sim \mathcal{N}(0, \sigma^2)$. Then the matrix $\mathbf{D}^{\mathbf{f}(\mathbf{x})} \in \mathbb{R}^{n\times k}$ that optimally decodes $\mathbf{f}(\mathbf{x})$ from the spike-trains $\boldsymbol{\delta}$ encoding $\mathbf{x}$ is obtained by solving the following problem (via regularized least-squares):

$$\mathbf{D}^{\mathbf{f}(\mathbf{x})} = \underset{\mathbf{D}\in\mathbb{R}^{n\times k}}{\arg\min} \int_S \left\| \mathbf{f}(\mathbf{x}) - \sum_{i=1}^{n} \left(r_i(\mathbf{x}) + \eta\right) \mathbf{d}_i \right\|^2 d^k\mathbf{x} \quad (3)$$

$$\implies \sum_{i=1}^{n} (\delta_i * h)(t)\mathbf{d}_i^{\mathbf{f}(\mathbf{x})} \approx (\mathbf{f}(\mathbf{x}) * h)(t). \quad (4)$$

The quantity in (4) may then be encoded via Principle 1 to complete the connection between two populations of neurons.[3]

C. Principle 3 – Dynamics

The third principle addresses the problem of implementing the following nonlinear dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{u}, \quad \mathbf{u}(t) \in \mathbb{R}^k$$
$$\mathbf{y} = \mathbf{g}(\mathbf{x}). \quad (5)$$

Since we take the synapse (2) to be the dominant source of dynamics for the represented vector [2, p. 327], we must essentially "convert" (5) into an equivalent system where the integrator is replaced by a first-order LPF. This transformation is accomplished by driving the filter $h(t)$ with:

$$\mathbf{w} := \tau\dot{\mathbf{x}} + \mathbf{x} = (\tau\mathbf{f}(\mathbf{x}) + \mathbf{x}) + (\tau\mathbf{u}) \quad (6)$$
$$\implies (\mathbf{w} * h)(t) = \mathbf{x}(t), \quad (7)$$

so that convolution with $h(t)$ achieves the desired integration. Therefore, the problem reduces to representing $\mathbf{x}(t)$ in a population of neurons using Principle 1, while recurrently decoding $\mathbf{w}(t)$ using the methods of Principle 2 (Fig. 1).

D. Extensions to Silicon Synapses

Consider an array of m heterogeneous pulse-extended second-order LPFs (in the Laplace domain):

$$H_j(s) = \frac{\gamma_j \left(1 - e^{-\epsilon_j s}\right) s^{-1}}{\left(\tau_{j,1}s + 1\right)\left(\tau_{j,2}s + 1\right)}, \quad j = 1 \ldots m, \quad (8)$$

where ϵ_j is the width of the extended pulse, γ_j is the height of the extended pulse, and $\tau_{j,1}$, $\tau_{j,2}$ are the two time-constants of the LPF. $H_j(s)$, whose circuit is described in §III, is an extended pulse $\gamma_j \left(1 - e^{-\epsilon_j s}\right) s^{-1}$ convolved with a second-order LPF $\left(\left(\tau_{j,1}s + 1\right)\left(\tau_{j,2}s + 1\right)\right)^{-1}$. These higher-order effects result in incorrect dynamics when the NEF is applied using the standard Principle 3 (e.g., in [5], [6], [7]), as shown in §IV.

[3]The effective weight-matrix in this case is $\boldsymbol{\mathcal{W}} = \mathbf{E}\left(\mathbf{D}^{\mathbf{f}(\mathbf{x})}\right)^T$.

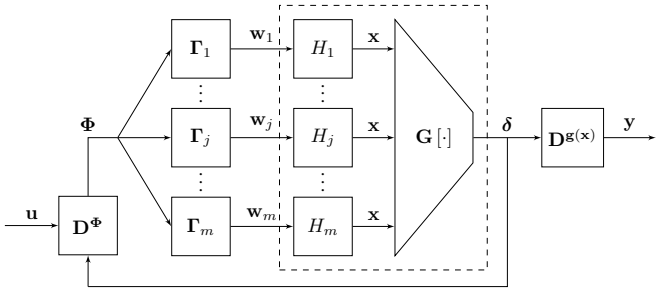

Fig. 2. Using extended Principle 3 (see (11)) to implement a general nonlinear dynamical system (see (5)) on a neuromorphic architecture. The matrix representation $\boldsymbol{\Phi}$ is linearly transformed by $\boldsymbol{\Gamma}_j$ to drive the j^{th} synapse. Dashed lines surround the silicon-neuron array that filters and encodes $\mathbf{x}$ into spike-trains (see Fig. 3).

From (7), observe that we must drive the j^{th} synapse with some signal $\mathbf{w}_j(t)$ that satisfies the following ($\mathbf{W}(s)$ denotes the Laplace transform of $\mathbf{w}(t)$ and we omit j for clarity):

$$\frac{\mathbf{X}(s)}{\mathbf{W}(s)} = H(s)$$
$$\iff \mathbf{W}(s)\ \left(1 - e^{-\epsilon s}\right) s^{-1}$$
$$= \gamma^{-1}(1 + (\tau_1 + \tau_2)s + \tau_1\tau_2 s^2)\mathbf{X}(s). \quad (9)$$

To solve for $\mathbf{w}$ in practice, we first substitute $1 - e^{-\epsilon s} = \epsilon s - (\epsilon^2 s^2)/2 + O(\epsilon^3 s^3)$, and then convert back to the time-domain to obtain the following approximation:

$$\mathbf{w} = (\epsilon\gamma)^{-1}(\mathbf{x} + (\tau_1 + \tau_2)\dot{\mathbf{x}} + \tau_1\tau_2\ddot{\mathbf{x}}) + \frac{\epsilon}{2}\dot{\mathbf{w}}. \quad (10)$$

Next, we differentiate both sides of (10):

$$\dot{\mathbf{w}} = (\epsilon\gamma)^{-1}(\dot{\mathbf{x}} + (\tau_1 + \tau_2)\ddot{\mathbf{x}} + \tau_1\tau_2\dddot{\mathbf{x}}) + \frac{\epsilon}{2}\ddot{\mathbf{w}},$$

and substitute this $\dot{\mathbf{w}}$ back into (10) to obtain:

$$\mathbf{w} = (\epsilon\gamma)^{-1}(\mathbf{x} + (\tau_1 + \tau_2 + \epsilon/2)\dot{\mathbf{x}} + (\tau_1\tau_2 + (\epsilon/2)(\tau_1 + \tau_2))\ddot{\mathbf{x}}) + (\epsilon\gamma)^{-1}(\epsilon/2)\tau_1\tau_2\dddot{\mathbf{x}} + (\epsilon^2/4)\ddot{\mathbf{w}}.$$

Finally, we make the approximation $(\epsilon\gamma)^{-1}(\epsilon/2)\tau_1\tau_2\dddot{\mathbf{x}} + (\epsilon^2/4)\ddot{\mathbf{w}} \ll \mathbf{w}$, which yields the following solution to (9):

$$\mathbf{w}_j = \boldsymbol{\Phi}\boldsymbol{\Gamma}_j, \quad (11)$$
$$\boldsymbol{\Phi} := \left[\mathbf{x}\ \dot{\mathbf{x}}\ \ddot{\mathbf{x}}\right],$$
$$\boldsymbol{\Gamma}_j := (\epsilon_j\gamma_j)^{-1} \begin{bmatrix} 1 \\ \tau_{j,1} + \tau_{j,2} + \epsilon_j/2 \\ \tau_{j,1}\tau_{j,2} + (\epsilon_j/2)(\tau_{j,1} + \tau_{j,2}) \end{bmatrix},$$

where $\epsilon_j\gamma_j$ is the area of the extended pulse.[4] We compute the time-varying matrix representation $\boldsymbol{\Phi}$ in the recurrent connection (plus an input transformation) via Principle 2 (similar

[4]This form for (6) is $\boldsymbol{\Phi} = [\mathbf{x}\ \dot{\mathbf{x}}]$ and $\boldsymbol{\Gamma}_j = [1,\ \tau]^T$. Further generalizations are explored in [12].

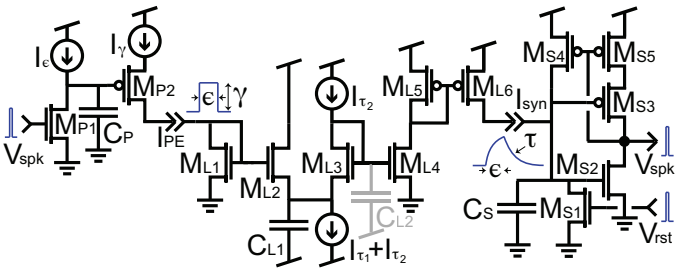

Fig. 3. Silicon synapse and soma. The synapse consists of a pulse-extender ($M_{P1,2}$) and a LPF ($M_{L1\text{-}6}$). The pulse-extender converts subnanosecond digital-pulses—representing input spikes—into submillisecond current-pulses (I_{PE}). Then the LPF filters these current-pulses to produce the synapse's output (I_{syn}). The soma integrates this output current on a capacitor (C_S) and generates a subnanosecond digital-pulse—representing an output spike—through positive-feedback ($M_{S2\text{-}5}$). This pulse is followed by a reset pulse, which discharges the capacitor. This schematic has been simplified for clarity.

to Fig. 1). We then drive the j^{th} synapse with its time-invariant linear transformation $\mathbf{\Gamma}_j$ (Fig. 2).

A more refined solution may be found by expanding $1 - e^{-\epsilon s}$ to the third order, which adds $\epsilon^2/12$ to the third coordinate of $\mathbf{\Gamma}_j$.[5] However, this refinement does not improve our results. Some remaining details are addressed in §IV.

III. Circuit Description

We now describe the silicon synapse and soma circuits (Fig. 3) that we use to validate our extensions to the NEF. An incoming pulse discharges C_P, which I_ϵ subsequently charges [9]. As a result, M_{P2} turns on momentarily, producing an output current-pulse I_{PE} with width and height:

$$\epsilon = C_P V_\gamma \,/\, I_\epsilon, \quad \gamma = I_\gamma,$$

where V_γ is the gate-voltage at which M_{P2} can no longer pass I_γ. This current-pulse is filtered to obtain the synapse's output, I_{syn}, whose dynamics obeys:

$$\tau_1 \frac{dI_{syn}}{dt} + I_{syn} = A\, I_{PE},$$

where $\tau_1 = C_{L1} U_T \,/\, I_{\tau_1}$, $A = I_{\tau_2} \,/\, I_{\tau_1}$, and U_T is the thermal voltage. The above assumes all transistors operate in the subthreshold region and ignores all parasitic capacitances [10]. If we include the parasitic capacitance C_{L2}, a small-signal analysis reveals second-order dynamics:

$$\tau_1 \tau_2 \frac{d^2 I_{syn}}{dt} + (\tau_1 + \tau_2) \frac{dI_{syn}}{dt} + I_{syn} = A\, I_{PE},$$

where $\tau_2 = C_{L2} U_T \,/\, \kappa I_{\tau_2}$, and κ is the subthreshold-slope coefficient. This second-order LPF and the aforementioned pulse-extender are modelled together in the Laplace domain by (8) after scaling by A.

The dynamics of the soma are described by:

$$\frac{C_S U_T}{\kappa I_m} \frac{dI_m}{dt} = I_m + I_{syn},$$

where I_m is the current in the positive-feedback loop, assuming all transistors operate in the subthreshold region [13], [14].

[5] In general, the coefficients 1, $\epsilon/2$, $\epsilon^2/12$, and so on, correspond to the $[0/q]$ Padé approximants of $\sum_{i=0}^{q} \frac{(-\epsilon)^i}{(i+1)!}$.

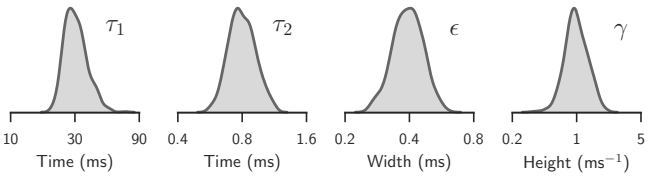

Fig. 4. Log-normally distributed parameters for the silicon synapses. τ_1 ($\mu \pm \sigma = 31 \pm 6.4$ ms) and τ_2 (0.8 ± 0.11 ms) are the two time-constants of the second-order LPF; ϵ (0.4 ± 0.06 ms) and γ ($1.0 \pm 0.29\,\text{ms}^{-1}$) are the widths and heights of the extended pulse, respectively (see (8)).

This equation may be solved to obtain the trajectory of I_m, and hence the steady-state spike-rate:

$$r(I_{syn}) = \frac{\kappa I_{syn}}{C_S U_T} \ln\left(\frac{I_{syn} \,/\, I_0 + 1}{I_{syn} \,/\, I_{thr} + 1}\right)^{-1},$$

where I_0 and I_{thr} are the values of I_m at reset and threshold, respectively. These values correspond to the leakage current of the transistors and the peak short-circuit current of the inverter, respectively.

IV. Validation with Circuit Models

We proceed by validating the extended NEF neuromorphic architecture (see Fig. 2) implemented using the circuit models (see §III) on two fundamental dynamical systems: an integrator and a controlled oscillator. We use Nengo 2.3.1 [4] to simulate this neuromorphic system with a time-step of $50\,\mu$s. Test data is sampled independently of the training data used to optimize (3) via regularized least-squares. For comparison with (5)—simulated via Euler's method—spike-trains are filtered using (2) with $\tau = 10$ ms. For each trial, the somatic parameters and synaptic parameters (Fig. 4) are randomly sampled from distributions generated using a model of transistor mismatch (validated in SPICE). These parameters determine each linear transformation $\mathbf{\Gamma}_j$, as defined in (11).

A. Integrator

Consider the one-dimensional integrator, $\dot{x} = u$, $y = x$. We represent the scalar $x(t)$ in a population of 512 modelled silicon neurons. To compute the recurrent function, we use (11), and assume $\ddot{x} = \dot{u}$ is given. To be specific, we optimize (3) for $\mathbf{D}^x$, and then add u and $\dot{u}$ to the second and third coordinates of the representation, respectively, to decode:

$$\mathbf{\Phi} = [x,\ u,\ \dot{u}]\,.$$

To evaluate the impact of each extension to the NEF, we simulate our network under five conditions: using standard Principle 3, accounting for second-order dynamics, accounting for the pulse-extender, accounting for the transistor mismatch in τ_1, and our full extension (Fig. 5). We find that the last reduces the error by 63%, relative to the first, across a wide range of input frequencies (5–50 Hz).

B. Controlled 2D Oscillator

Consider a controllable two-dimensional oscillator,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{u}, \quad \mathbf{y} = \mathbf{x},$$
$$\mathbf{f}(x_1, x_2, x_3) = [-\omega x_3 x_2,\ \omega x_3 x_1,\ 0]^T,$$

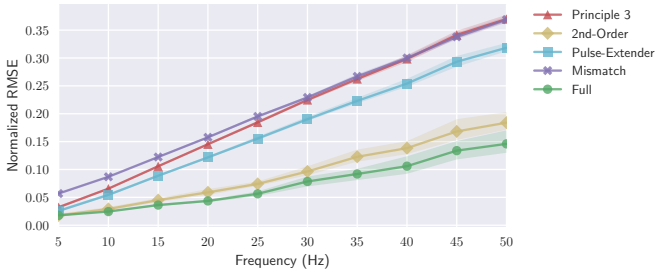

Fig. 5. Effect of each NEF extension, applied to a simulated integrator given frequencies ranging from 5–50 Hz (mean spike-rate 143 Hz). The standard approach (Principle 3) achieves a normalized RMSE of 0.203 with 95% CI of $[0.189,\ 0.216]$ compared to the ideal, while our extension (Full) achieves 0.073 with 95% CI of $[0.067,\ 0.080]$—a 63% reduction in error—averaged across 25 trials and 10 frequencies. The largest improvement comes from accounting for the second-order dynamics, while a smaller improvement comes from accounting for the pulse-extended dynamics. Accounting for the transistor mismatch on its own is counter-productive.

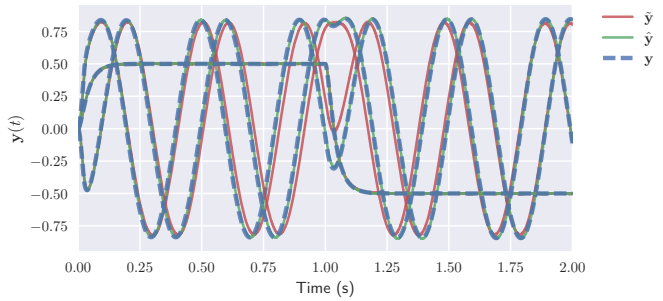

Fig. 6. Output of controlled 2D oscillator with $\omega = 5$ Hz (mean spike-rate 140 Hz). The control (x_3^*) is changed from 0.5 to -0.5 at 1 s to reverse the direction of oscillation. The standard Principle 3 ($\tilde{\mathbf{y}}$) achieves a normalized RMSE of 0.188 with 95% CI of $[0.166,\ 0.210]$ compared to the ideal ($\mathbf{y}$), while our extension ($\hat{\mathbf{y}}$) achieves 0.050 with 95% CI of $[0.040,\ 0.063]$—a 73% reduction in error—averaged across 25 trials.

where ω is the angular frequency in radians per second, x_3 controls this frequency multiplicatively, and x_3^* is the fixed-point target supplied via input $u_3 = x_3^* - x_3$. The inputs u_1 and u_2 initiate the oscillation with a brief impulse. We represent the three-dimensional state-vector $\mathbf{x}(t)$ in a population of 2048 modelled silicon neurons.[6] To compute the recurrent function, we again use (11). For this example, $\mathbf{u}(t) \approx \mathbf{0}$ for most t (apart from initial transients and changes to the target x_3^*), and so $\ddot{\mathbf{x}} = \mathbf{J_f}(\mathbf{x}) \cdot \dot{\mathbf{x}} + \dot{\mathbf{u}} \approx \mathbf{J_f}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$ (where $\mathbf{J_f}$ denotes the Jacobian of $\mathbf{f}$). We then optimize (3) for $\mathbf{D^x}$, $\mathbf{D^{f(x)}}$, and $\mathbf{D^{J_f(x)\cdot f(x)}}$, and add $\mathbf{u}$ to the second column of the matrix representation to decode:

$$\mathbf{\Phi} = [\mathbf{x},\ \mathbf{f}(\mathbf{x}) + \mathbf{u},\ \mathbf{J_f}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})]\,.$$

We find that this solution reduces the error by 73% relative to the standard Principle 3 solution (Fig. 6).

V. Summary

We have provided a novel extension to the NEF that directly harnesses the dynamics of heterogeneous pulse-extended second-order LPFs. This theory is validated by software simulation of a neuromorphic system, using circuit models with parameter variability validated in SPICE, for two fundamental examples: an integrator and a controlled oscillator. When compared to the previous standard approach, our extension is shown to reduce the error by 63% and 73% for the integrator and oscillator, respectively. Thus, our theory enables a more accurate mapping of nonlinear dynamical systems onto a recurrently connected neuromorphic architecture using non-ideal silicon synapses. Furthermore, we derive our theory independently of the particular neuron model and encoding. This advance helps pave the way toward understanding how non-ideal physical primitives may be systematically analyzed and then subsequently exploited to support useful computations in neuromorphic hardware.

[6] We use this many neurons in order to minimize the noise from spiking.

Acknowledgements

This work was supported by CFI and OIT infrastructure, the Canada Research Chairs program, NSERC Discovery grant 261453, ONR grants N000141310419 and N000141512827, and NSERC CGS-D funding. The authors thank Wilten Nicola for inspiring (9) with a derivation for the double-exponential.

References

[1] C. Mead, *Analog VLSI and Neural Systems*. Boston, MA: Addison-Wesley, 1989.

[2] C. Eliasmith and C. H. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT Press, 2003.

[3] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[4] T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. C. Stewart, D. Rasmussen, X. Choo, A. R. Voelker, and C. Eliasmith, "Nengo: A Python tool for building large-scale functional brain models," *Frontiers in neuroinformatics*, vol. 7, 2013.

[5] S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen, "Silicon neurons that compute," in *International Conference on Artificial Neural Networks*, vol. 7552. Springer, 2012, pp. 121–128.

[6] S. Menon, S. Fok, A. Neckar, O. Khatib, and K. Boahen, "Controlling articulated robots in task-space with spiking silicon neurons," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, 2014, pp. 181–186.

[7] F. Corradi, C. Eliasmith, and G. Indiveri, "Mapping arbitrary mathematical functions and dynamical systems to neuromorphic VLSI circuits for spike-based neural computation," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 269–272.

[8] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski, "Synaptic currents, neuromodulation and kinetic models," *The handbook of brain theory and neural networks*, vol. 66, pp. 617–648, 1995.

[9] J. V. Arthur and K. Boahen, "Recurrently connected silicon neurons with active dendrites for one-shot learning," in *International Joint Conference on Neural Networks (IJCNN)*, vol. 3. IEEE, 2004, pp. 1699–1704.

[10] W. Himmelbauer and A. G. Andreou, "Log-domain circuits in subthreshold MOS," in *Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium on*, vol. 1. IEEE, 1997, pp. 26–30.

[11] K. E. Friedl, A. R. Voelker, A. Peer, and C. Eliasmith, "Human-inspired neurorobotic system for classifying surface textures by touch," *Robotics and Automation Letters*, vol. 1, no. 1, pp. 516–523, 01 2016.

[12] A. R. Voelker and C. Eliasmith, "Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells," 2017, Manuscript in preparation.

[13] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "A biomorphic digital image sensor," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, 2003.

[14] P. Gao, B. V. Benjamin, and K. Boahen, "Dynamical system guided mapping of quantitative neuronal models onto neuromorphic hardware," *IEEE Transactions on Circuits and Systems*, vol. 59, no. 10, pp. 2383–2394, 2012.