Published in final edited form as:

IEEE Trans Biomed Circuits Syst. 2012 April ; 6(2): 133–141. doi:10.1109/TBCAS.2011.2182650.

Parameter Estimation of a Spiking Silicon Neuron

Alexander Russell,

Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA

Kevin Mazurek [Student Member, IEEE],

Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA

Stefan Mihalas,

Department of Neuroscience and the Zanvyl Krieger Mind Brain Institute, The Johns Hopkins University, Baltimore, MD 21218 USA

Ernst Niebur, and

Department of Neuroscience and the Zanvyl Krieger Mind Brain Institute, The Johns Hopkins University, Baltimore, MD 21218 USA

Ralph Etienne-Cummings [Senior Member, IEEE]

Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA

Alexander Russell: alexrussell@jhu.edu; Kevin Mazurek: kmazurek@jhu.edu; Stefan Mihalas: mihalas@jhu.edu; Ernst Niebur: niebur@jhu.edu; Ralph Etienne-Cummings: reti-enne@jhu.edu

Abstract

Spiking neuron models are used in a multitude of tasks ranging from understanding neural behavior at its most basic level to neuroprosthetics. Parameter estimation of a single neuron model, such that the model's output matches that of a biological neuron is an extremely important task. Hand tuning of parameters to obtain such behaviors is a difficult and time consuming process. This is further complicated when the neuron is instantiated in silicon (an attractive medium in which to implement these models) as fabrication imperfections make the task of parameter configuration more complex. In this paper we show two methods to automate the configuration of a silicon (hardware) neuron's parameters. First, we show how a Maximum Likelihood method can be applied to a leaky integrate and fire silicon neuron with spike induced currents to fit the neuron's output to desired spike times. We then show how a distance based method which approximates the negative log likelihood of the lognormal distribution can also be used to tune the neuron's parameters. We conclude that the distance based method is better suited for parameter configuration of silicon neurons due to its superior optimization speed.

Index Terms

Neuromorphic; parameter estimation; silicon neuron

I. Introduction

Over 100 years ago, Llapique [1] introduced the Integrate and Fire neuron model. Since then, multiple neuron models have evolved, finding uses in tasks ranging from understanding information processing in the brain [2], providing greater understanding of the mechanoreceptive afferent fibers [3] as well as the neural coding of sound [4], to designing the next generation of neuroprosthetics [5], bipedal locomotion controllers [6] and image recognition systems [7]. The scientific question to be answered, or the engineering task at hand, dictates the choice of neuron model to be used. If the aim is to understand the neuron at the most basic level then models such as the Hodgkin–Huxley [8] and Morris–Lecar [9] models should be used. These models incorporate the dynamics of voltage-gated ion channels within the neuron’s membrane, using complicated sets of equations, to explicitly account for the spike generation mechanism of the neuron. Such models provide insights into how neurons operate at their most fundamental level.

The biological accuracy of the above models comes at the cost of implementation efficiency (i.e., the number of FLOPS necessary to implement the model in software [10] or the circuit complexity and silicon real-estate to implement the model in hardware [11]), and large parameter spaces. It is possible to trade biological accuracy for efficiency by emulating the function of the neuron without attempting to accurately represent the biological mechanism of the neuron [12]. For example, Izhikevich [13] proposed a model which is able to reproduce almost all modes of cortical spiking in an exceptionally compact manner. The model is, however, a mathematical fit of a neuron’s dynamics and consequently does not provide any biological insights to be made through its parameters. Alternatively, models can be based around the Leaky Integrate and Fire (LIF) neuron proposed by Llapique [1]. These models achieve efficiency by disregarding the spike generation process whilst maintaining subthreshold dynamics [14]. Furthermore, direct parallels can be made between the model’s parameters and the properties of a biological neuron.

Although the basic LIF neuron has very few spiking modalities, it is easily extended to include a rich repertoire of spiking modalities by the introduction of spike-induced currents [15]. The work reported in this paper makes use of this style of neuron; however it should be mentioned that spike induced currents alone are not capable of allowing for all known modes of spiking. If behaviors such as bursting with adaptation or anode-break spiking—the phenomena where some neurons spike after the removal of a hyperpolarizing current, that has been applied for an extended period without any injection of a depolarizing current—are desired then a variable threshold could be added [16].

An important task for both neuroscientists and engineers is determining the best way to estimate the parameters of a spiking neuron model such that it outputs a specified behavior. Much work has been done on configuration methods for use with software neuron models (see Fig. 1) [17]–[21]; however it is often desirable to implement these neural models in silicon¹ as it allows for high density, low power, realtime networks of spiking neurons to be instantiated. This further complicates the parameter estimation process as transistor mismatch, fabrication irregularities, non-ideal current sources and transistor noise will result in the *in silico* instantiation of the model not being identical to the simulated system. Consequently, the parameters which worked in simulation will not necessarily work in hardware. Methods to automate this mapping are an area of active research. Neftci *et al.* [22] developed a method to map software parameters to silicon neurons using only the neuron’s firing rate as the optimization metric. However this method proved to be inaccurate for

¹The term software model is used to describe a neuron model implemented in software and the terms silicon or *in silico* are used to describe a neuron model instantiated in custom analog VLSI hardware.

single neurons and was better suited for mapping the parameters of neural populations where the errors associated with individual neurons average out. Bruederle *et al.* [23] use the neurons' state variables to design the mapping algorithm. However in an integrated circuit containing multiple neurons, it is unlikely that there is access to the state variables of every neuron on the chip. Consequently circuit simulations, as in the case of Bruederle *et al.*, need to be used to design the mapping. Simulations cannot exactly predict the imperfections and irregularities which will affect the fabricated neurons. Like the method used by Neftci *et al.*, simulation based mapping methods are thus better suited for neural populations where errors average out. However, we are interested in setting the parameters of a hardware neuron such that it exhibits a specified behavior. A method is either needed to accurately map the parameters of a software neuron (whose parameters have already been optimized such that it exhibits the correct firing times) to a hardware neuron, or to estimate the parameters of the hardware neuron directly. Saighi *et al.* [24] use a voltage clamp method to calibrate the channel conductances of a silicon Hodgkin–Huxley model. Using this technique they are able to calibrate the silicon neuron's parameters such that it exhibits spiking behaviors (fast and slow spiking modalities) qualitatively similar to those observed in a biological neuron. However, the patch clamp method does not allow for neural parameters to be found such that the neuron's spike times match those of a desired action potential pattern. Furthermore, this method requires knowledge of the internal dynamics of both the silicon neuron being configured and the biological neuron being mimicked.

In the remainder of the paper we demonstrate two optimization methods—the Maximum Likelihood method and Time Ratio Distance method—which can be used to directly find the parameters of a silicon neuron such that it outputs a specified spike pattern. Both the Maximum Likelihood method and Time Ratio Distance methods require only the spike outputs of the neuron being optimized. Consequently these methods are compatible with any integrate and fire based neuromorphic hardware systems which have spike outputs.

II. The Optimization Problem

Given a set of n spike times, $t_1, \dots, t_n$, we wish to build a neural model which can accurately predict them. The first step of this process is choosing a model which can capture all the necessary spiking modalities. For instance, if bursting is required then the model must provide mechanisms to allow for this—e.g., through the inclusion of spike induced currents. For the work presented in this paper, the Noisy Leaky Integrate and Fire (NLIF) model with spike induced currents, described in Section III, is used.

Now that a model structure has been chosen the next step is to find the parameters $\mathbf{X}$ of the model such that it is able to accurately predict the desired spike times. The higher the cardinality of $\mathbf{X}$, the slower the optimization process. For efficient parameter tuning, assumptions, such as setting $V_{\text{reset}} = V_{\text{leak}}$, in (4), should be made to reduce the free parameters of the model. Furthermore, if it is available, existing knowledge of the neuron to be emulated should be utilized. If for example, the biological neuron being modeled contained an NR2B glutamergic NMDA channel then the time constant of the model's spike induced current could be set to 500 ms [19].

In order to perform the optimization, a metric or cost function, that describes the model's performance is needed. Cost functions for spiking neurons can roughly be placed into two main classes: distance measures and likelihood measures. Distance based measures have the advantage that they are computationally efficient; however most of these measures, such as those in [25]–[27], add free parameters to the optimization process (for example the spike distance measure of Victor and Purpura [25] requires 3 additional parameters: the cost of

adding a spike, the cost of removing a spike and the cost of moving a spike). Furthermore, these metrics do not guarantee a global minimum [19].

An alternative method is to compute the likelihood of a spike train occurring given a model and its parameters. If the i th inter-spike interval starts at t_{i-1} and ends at t_i , then the conditional probability density of an action potential occurring during this interval can be defined as

$$f_i(t|t_{i-1}, \dots, t_1). \quad (1)$$

This is also the likelihood of a spike occurring at time t during the i th interspike interval. The likelihood for the entire spike train can then be written using the chain rule from probability as

$$Li(t_n, \dots, t_1) = \prod_{i=1}^n f_i(t_i|t_{i-1}, \dots, t_1). \quad (2)$$

If the logarithm of this is taken then the negative log-likelihood of the entire spike train can be written as

$$Li_{\log}(t_n, \dots, t_1) = - \left(\sum_{i=1}^n \log(f_i(t_i|t_{i-1}, \dots, t_1)) \right). \quad (3)$$

Paninski *et al.* [17] showed how the likelihood of a spike train generated using the NLIF neuron could be calculated if the noise in the membrane potential is Gaussian. Furthermore, they showed that the negative log-likelihood was concave in shape. Consequently, the negative log likelihood has a unique minimum allowing for conventional optimization techniques to be used. Likelihood based cost functions also have the advantage in that they do not add any additional parameters to the optimization process. In the section that follows we review the basic theory of the Maximum Likelihood method developed by [17] and then test the method for use with silicon neurons by configuring the *in silico* NLIF neuron for bursting. Some of this work (specifically the Maximum Likelihood method) has previously been published in [28]. Next we show how the negative log likelihood can be approximated using the Time Ratio Distance metric, developed by Mihala *et al.* [21]. This is a distance based metric that approximates the negative log likelihood of the lognormal distribution. This cost function does not introduce any free parameters to the optimization process and combines the efficiency of distance based methods with the functionality of likelihood based methods [21].

III. The Noisy Leaky Integrate and Fire Neuron

The dynamics of Noisy Leaky Integrate and Fire (NLIF) Neuron with spike induced currents used in this work are as follows:

$$dV(t) = \frac{1}{C} (-g(V(t) - V_{\text{leak}}) + I_{\text{ext}}(t) + I_{\text{ps}}(t)) dt + N_t \quad (4)$$

where V is the neuron's membrane potential, g is the leak conductance, C the membrane capacitance, V_{leak} the extra cellular potential, I_{ext} is the external stimulus current applied to the neuron, I_{ps} is a spike induced current and N_t is noise. I_{ext} can be an input signal projected onto a linear filter, however in this paper we consider a simpler case where it is simply a constant. The noise term, N_t , consists of all noise sources intrinsic to the silicon instantiation

of the NLIF (such as thermal and flicker noise) as well any noise injected into the neuron's membrane (see Section IV). The implementation of the spike induced current, I_{ps} , differed for the Maximum Likelihood algorithm and the Time Ratio Distance algorithm. Details of I_{ps} are given below. When $V(t)$ reaches the neuron's threshold, V_{thresh} , then a spike occurs and the neuron resets according to

$$V \leftarrow V_{\text{reset}}. \quad (5)$$

The addition of spike induced currents ensures that the NLIF neuron is able to assume many different firing modalities. If more complicated behaviors were required, the NLIF could easily be extended by adding a filter to transform an input stimulus to a current [5], [17] or by introducing a variable threshold to allow for anode break spiking or both adaptation and bursting [16].

A. Spike Induced Current Generation

Two different methods of generating the spike induced current, I_{ps} , were used in this work. For the Maximum Likelihood experiment, the current is described by

$$I_{ps}(t) = \sum_{j=0}^{i-1} h(t-t_j) \quad (6)$$

where h is a postspike current waveform of fixed amplitude and shape. The value of h depends only on the time since the last spike t_{i-1} and the sum includes terms back to t_0 —the first observed spike. $h(t)$ is constructed using raised cosine basis functions with a logarithmic stretching of time given by [17]

$$\text{hbasis}_i(t) = \frac{\cos(\max(-\pi, \min(\pi, \log(t)-c)\frac{2\pi}{dc})) + 1}{2} \quad (7)$$

where t is the time since the last spike, dc the spacing between the raised cosine peaks and c the centers of the basis vectors. The basis functions are combined to form $h(t)$ as follows:

$$h(t) = A_1 \times \text{hbasis}_1(t) + A_2 \times \text{hbasis}_2(t). \quad (8)$$

Only two basis functions were used as they are sufficient to cause bursting in the NLIF neuron; however, the number of basis functions can be arbitrary large.

For the Time Ratio Distance algorithm a slightly simpler current generation method was used where

$$I_{ps} = I_1 + I_2 \quad (9)$$

where I_1 is an excitatory current and I_2 is an inhibitory current. These currents were generated as follows [16]:

$$I'_j(t) = -k_j I_j, \quad j=1, 2 \quad (10)$$

where k_j is the current's time constant. When a spike occurs at time t_j the currents are updated according to

$$I_j(t_i^+) \leftarrow R_j \times I_j(t_i^-) + A_j, \quad j=1, 2 \quad (11)$$

where t_i^- and t_i^+ are the times immediately before and after a spike, R_j describes the history dependence of the current on its value before a spike, and A_j describes the independent component of the current's update. For example if $R_j = 0$ then I_j is independent of its final value during the previous interspike interval.

B. Hardware Implementation

The circuitry for the *in silico* NLIF neuron was based on that by Tenore *et al.* [29] and implemented in a 0.5 μm process. A leakage channel was added using a switched-capacitor circuit for an easily controllable channel conductance given by

$$g = C_s f_s \quad (12)$$

where g is the channel conductance, C_s is the capacitor value and f_s is the switching frequency. To ensure maximum flexibility and allow for spike induced currents of any shape and time scale, the spike induced currents are generated off chip using a Mathworks (Mathworks, Natick, MA) Real-Time PC (xPC). A National Instruments PCI6040e DAQ card was used to convert the current profiles to analogue voltages for interfacing with the NLIF chip.

IV. Maximum Likelihood Estimation

Given a set of spike times $t_1, t_2, \dots, t_n$ we wish to find the parameters $\mathbf{X} = \{V_{\text{reset}}, I_{\text{ext}}, A_1, A_2\}$ which cause the NLIF neuron to spike at the desired times (to reduce the parameter space we held g , V_{thresh} and the parameters for the shape of the raised cosine basis constant and set $V_{\text{resrt}} = V_{\text{leak}}$). Paninski *et al.* [17] made use of the fact that the NLIF neuron's subthreshold dynamics were linear to develop a likelihood function of the neuron firing at the correct times. It was proven that the negative log-likelihood function is concave and consequently has a unique minimum which allows for traditional optimization techniques to be used. This is presented in depth in [17] and so only a very brief overview of the necessary equations are given here.

The NLIF model's noiseless membrane potential can be analytically solved to be

$$V(t) = V_{\text{leak}} + (V_{\text{reset}} - V_{\text{leak}})e^{-\frac{g}{C}(t-t_{i-1})} + \frac{I_{\text{ext}}(t) + I_{\text{ps}}(t)}{C} * e^{-\frac{g}{C}(t-t_{i-1})} \quad (13)$$

where $*$ denotes convolution. The above has linear dynamics, consequently, when noise is added to the process [see (4)] then the membrane assumes the distribution of the noise. The noise term $N(t)$ can be written as

$$N(t) = W(t) + \eta(t) \quad (14)$$

where $W(t)$ is Gaussian white noise injected into the membrane and $\eta(t)$ is a result of all noise processes inherent to the silicon neuron. Setting $W(t)$ such that

$$W(t) \gg \eta(t) \quad (15)$$

then

$$N(t) \approx W(t) \quad (16)$$

and the membrane potential assumes a Gaussian probability density which we denote as $G(V(t) | \mathbf{X})$. Equation (13) is the mean of the distribution and its covariance can be calculated to be

$$\text{COV}(t_1, t_2) = \frac{1}{2g} (e^{-g|t_2-t_1|} - e^{-g(t_1+t_2)}). \quad (17)$$

On a given interspike interval $[t_{i-1}, t_i]$, the set

$$C_i = \bigcap_{t_{i-1} \leq t < t_i} \{V(t) < V_{\text{reset}}\} \cap \{V(t_i) \geq V_{\text{reset}}\} \quad (18)$$

describes the set of all possible paths that the membrane potential may take such that an action potential occurs at time t_i . The likelihood that the neuron spikes at time t_i given that there was a spike at time t_{i-1} is the probability of the event $V(t) \in C_i$ which is given by

$$\int_{V(t) \in C_i} G(V(t) | \mathbf{X}). \quad (19)$$

When a spike occurs then the membrane potential resets to V_{reset} . The noise contribution by N_t is independent between interspike intervals. The system is thus a first-order nonhomogeneous Markov process and the likelihood of the entire spike train can be written as the multiplication of the conditional probabilities for each individual spike in the train [30].

$$Li(t_n, \dots, t_1) = \prod_i \int_{V(t) \in C_i} G(V(t) | \mathbf{X}) \quad (20)$$

A. Computation

An efficient method is needed to calculate the likelihoods given in (20). Paninski *et al.* used the fact that the probability evolution of the membrane potential from t_{i-1} to t_i satisfies the Fokker-Planck equation [17], [31] to compute the likelihoods. This is a partial differential equation not easily solved in hardware. Instead, we calculate the likelihood equation using Monte Carlo methods as follows: Let $P(V, t)$ be the probability of the membrane potential being below threshold for a given voltage path and time. Then $\int P(V, t) dV$ is the probability that the neuron has not yet spiked at time t , given that the last spike was at time t_{i-1} . Thus $1 - \int P(V, t) dV$ is the cumulative distribution of the spike occurring at time t . Therefore

$$f(t) = -\frac{\partial}{\partial t} \int P(V, t) dV \quad (21)$$

is the conditional probability density of a spike at time t . Now if we discretize the interspike interval into j bins each of width Δt then for any bin k we can calculate

$$\int P(V, k\Delta t) dV = \frac{m_k}{n} \quad (22)$$

where m_k is the number of iterations that have yet to spike by bin k and n is the total number of iterations used in the Monte Carlo simulation. Using this we can then define

$$f(k\Delta t) = -\frac{m_k - m_{k-1}}{n\Delta t} \quad (23)$$

which is the likelihood of a spike occurring in bin k . The log-likelihood is computed by taking the logarithm of (23) and the log likelihood for the entire spike train is obtained by adding the individual log-likelihoods together as shown in (25).

B. Configuring the NLIF Neuron for Bursting

The ML algorithm was tested by using it to find parameters to cause the neuron to burst at predetermined intervals. First the neuron's parameters were hand tuned to find a spiking pattern of interest—in this case a rhythmic bursting pattern. The interspike intervals (ISI's) of the burst were recorded and the following parameters were optimized $\mathbf{X} = \{V_{\text{reset}}, I_{\text{ext}}, A1, A2\}$. The initial values of $A1$ and $A2$ were set to zero and the remaining parameters were randomly assigned to fall on either the high or low boundary of their normal operating range. This range was chosen such that all transistors were kept in their desired operating modes as determined through schematic simulation. This ensured that the initial condition resulted in the neuron spiking at a rate far away from that observed in the desired bursting pattern and with no spike induced currents. The number of simulations per Monte Carlo trial was set to 100 and the simplex based MATLAB (Mathworks, Natick, MA) algorithm *fminsearch* was used to traverse the parameter space.

To ensure that the log likelihood function is smooth, and contains no local minima due to insufficient Monte Carlo simulations, the envelope of the likelihood calculated from the Monte Carlo simulations was used. This allowed a relatively small number of trials to be used, substantially speeding up the algorithm. To help convergence, a coarse to fine optimization strategy was implemented where both the width of the time bins and variance of the noise decreased as the optimization progressed.

V. Spike Time Ratio Distance (TRD) Optimization

The Maximum Likelihood method discussed in the previous section is very slow to compute due to the large number of iterations needed to compute the likelihood. It has been observed in biological neurons, that the likelihood of the neuron firing at a given time closely follows the lognormal distribution [32]–[35]. Fig. 2 shows the likelihood of the NLIF neuron computed over 30 000 Monte Carlos simulations along with the best fit normal distribution and lognormal distribution. It can be seen that, like in biology, the lognormal distribution closely approximates the NLIF's likelihood. Massive time savings could be made if the likelihood of the neuron is assumed to follow the lognormal distribution. Mihala *et al.* [21] took advantage of this fact in designing a cost function for use with the Mihalas-Niebur neuron [16] whose likelihood also closely resembles a lognormal distribution. They investigated various approximations of the negative log-likelihood and developed a very efficient, distance based metric called the Time Ratio Distance (TRD) metric which combines the efficiency of distance based metrics with the power of the likelihood based metric. By designing the TRD cost function to mimic the likelihood function in the ML method, it is expected to have no local minima; however there is no guarantee of this. The TRD approximates the negative log likelihood of the lognormal distribution to the second order and is defined as follows:

$$\text{TRD}(t_i, t'_i) = \left(1 - \frac{\min(t_i, t'_i)}{\max(t_i, t'_i)}\right)^2 \quad (24)$$

where t'_i is the desired spike time and t_i the actual spike. For details of the derivation of this function see [21]. The negative log likelihood of a spike train is then calculated by using the TRD approximation in (25) as follows:

$$Li_{\text{TRD}}(t_n, \dots, t_1) = \sum_{i=1}^n \text{TRD}(t_i, t'_i). \quad (25)$$

A. Configuring the NLIF Neuron for Bursting

To test the TRD method a very similar experiment to that used to test the Maximum Likelihood algorithm was conducted. Again the neuron's parameters were hand tuned so that the model exhibits a rhythmic bursting pattern. The ISI's of the burst were recorded and used as the target ISI's. In this experiment the parameters $\mathbf{X} = \{V_{\text{reset}}, I_{\text{ext}}, A1, A2, R1, R2\}$ were optimized. The initial values of $A1, A2, R1, R2$ were all set to zero and the remaining parameters were randomly assigned to fall on either the high or low boundary of their normal operating range. This range was chosen such that all transistors were kept in their desired operating modes as determined through schematic simulation. This ensured that the initial condition resulted in the neuron spiking at a rate far away from that observed in the desired bursting pattern and with no spike induced currents. As discussed in Section II, where possible, existing knowledge of the neuron to be modeled should be taken into consideration. The synaptic time constants are often known and so the time constants of the spike induced currents were not added as free parameters to the model. The time constant of the excitatory spike induced current, I_1 , was set to 10 ms and the time constant of the inhibitory spike induced current, I_2 , was set to 50 ms.

As shown in (4), the silicon neuron is intrinsically noisy. Consequently, there is a chance that parameters which on average are a good solution for the model may appear as a bad solution. Similarly, there is a chance that parameters which on average are a bad solution for the model may appear as a good solution. Two steps were taken to avoid the noise induced local minima in the optimization process. Firstly, each ISI was simulated 5 times and the average ISI was used to calculate the TRD. Secondly, a stochastic simplex based optimization algorithm called SIMPSA (Simplex-Simulated Annealing) was used. This algorithm is described in Cardoso *et al.* [36] and can be downloaded from <http://biomath.ugent.be/brecht/download.html>. The stochasticity of the algorithm allows it to escape local minima.

VI. Results

A. Maximum Likelihood

The results of the Maximum Likelihood parameter estimation are shown in Fig. 3. The desired spike times are shown in the top row and the predicted spike times are shown below where each row represents a new trial. It can be seen that the predicted spike times match the desired spike times almost identically and on average (i.e., across all 10 trials) the predicted spike trains are within 4% of the desired spike trains.

Fig. 4(a) shows a set of Monte Carlo simulations recorded towards the end of the optimization process. The density of the dots can be interpreted as the probability of the membrane potential assuming that value at a particular time. Note the increase in variance of

probability over time as predicted by (17) and the collapsing of the probability when the neuron is reset. The targeted interspike intervals are labeled 1,2,3 and 4 in the figure and the likelihood of spiking during these intervals is shown in Fig. 4(b).

Fig. 5 shows a histogram of the interspike intervals recorded from the optimized neuron over 100s. The peaks labeled and correspond to the targeted interspike intervals demarcated 1', 2', 3' and 4' in Fig. 4(a) and can be considered to be the mean values of the optimized interspike intervals. The variance around the peaks is a result of the noise inherent to the system. ISIs 1–3 are all very similar in length (27, 25 and 31 ms, respectively), consequently the jitter in the system results in a time overlap of ISI 2 and 3 with ISI 1. This results in the peak labeled 1' having a much higher spike count than any of the other peaks.

B. Time Ratio Distance

The results of the Time Ratio Distance parameter estimation are shown in Fig. 6. The top row of the raster plot shows the desired spike times. The predicted spike times for each of the repeated trials are shown below this. The variability across trials for the predicted spike trains is a result of noise within the system. Once again the predicted spike times closely match the desired spike times and on average (i.e., across all 10 trials) the predicted spike trains are within 3% of the desired spike trains.

VII. Discussion

The results show that both the Maximum Likelihood and Time Ratio Distance parameter estimation methods work with a silicon neuron. The optimized ISIs were within 4% and 3% (maximum error) of the desired ISIs for the ML method and TRD method respectively. This figure was limited by the time resolution of our system which was 1 ms. If a faster data acquisition system was used then smaller time bins could have been used and the error reduced. The Time Ratio Distance was substantially quicker (over an order of magnitude) than the Maximum Likelihood method as only 5 simulations of a spike train were needed to test a parameter set as opposed to 100 as in the ML case. Furthermore, the TRD method converged in significantly fewer iterations of the optimization process than the ML method. The speed up would be further enhanced if a lower noise system were used so that averaging of the ISIs for the TRD method was not necessary (or inversely a greater relative speed up would occur if more simulations were generated to build a more accurate likelihood for the ML method).

Which is the best method for parameter estimation? The TRD method is susceptible to local minima in high noise scenarios. If the TRD measure were to be applied in such situations then it is necessary to simulate the predicted spike train multiple times in order to get an accurate average of the ISIs. This can slow the TRD algorithm substantially so that its speed increase over the ML method is not so apparent. In such a situation it may be more beneficial to use the ML method (if the noise can be approximated as Gaussian) as Paninski *et al.* [17] have shown that there is only a single global minimum for the negative log likelihood (In such a scenario the TRD method could be used, with limited averaging, to provide a quick estimate of initial conditions for the ML algorithm. The ML algorithm could then be used to find the optimal parameters).

In other situations the TRD method is preferable, not only because of its computational efficiency but because it is more flexible. The Maximum Likelihood method requires a neuron with linear dynamics and the log-concavity of its likelihood has only been proven for the case where the noise is Gaussian. Many neurons, however, have more complex dynamics exhibiting behaviors such as saturation and adaptation. Successful emulation of these neurons will require more complex models that can account for such spiking

phenomenologies. Maximum Likelihood methods are not guaranteed to work in such cases; however the TRD approach can still be successfully applied [21]. Both ML and TRD are however better suited for the parameter estimation of silicon neurons than distance based measures such as the point process transformation [25], squared integral measure [26] or the coincidence factor [27] which have been found to have multiple local minima and therefore very rarely converge to the correct solution [21]. Furthermore, these distance measures complicate the parameter estimation process through the introduction of free parameters.

If a software implementation of the desired model exists then it may be possible to directly map its parameters to the silicon neuron to obtain the desired result. This; however, would require that the transistors making up the silicon neuron need to be individually characterized. This is extremely time consuming and most likely impossible for integrated circuits containing multiple neurons where only the spike outputs are available. A better approach would be to use software defined parameters as an educated initial condition for the ML or TRD algorithms thereby aiding convergence of the algorithm.

VIII. Conclusion

An *in silico* neuron which utilizes Maximum Likelihood or Time Ratio Distances to set its action potential times has been designed. The maximum likelihood algorithm developed by Paninski *et al.* and the Time Ratio Distance algorithm developed by Mihala *et al.* were both successfully used to tune the parameters of the chip to allow for bursting. The Time Ratio Distance is significantly faster than the Maximum Likelihood method and is thus preferable for the parameter estimation of *in silico* spiking neurons.

Acknowledgments

This work was supported by the Office of Naval Research under MURI Grant N000141010278 and NIH-NEI 5R01EY016281-02.

References

1. Lapicque L. Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *Journal de Physiologie et Pathologie Generale*. 1907; 9:620–623.
2. Sidiropoulou K, Kyriaki P, Panayiotou P. Inside the brain of a neuron. *Eur Mol Biol Org*. 2006; 7(9): 886–892.
3. Kim S, Sripati A, Bensmaia S. Predicting the timing of spikes evoked by tactile stimulation of the hand. *J Neurophys*. 2010; 104:1484–1496.
4. Michel, C.; Nouvian, R.; Azevedo-Coste, C.; Puel, J.; Bourien, J. A computational model of the primary auditory neuron activity. *Proc. IEEE Engineering Medicine and Biology Society*; Buenos Aires, Argentina. Sep. 2010; p. 722-725.
5. Bensmaia, S.; Kim, SS.; Sripati, A.; Vogelstein, RJ. Conveying tactile feedback using a model of mechanotransduction. *Proc. IEEE Conf. Biomedical Circuits and Systems*; Baltimore, MD. Nov. 2008; p. 137-140.
6. Lewis, M.; Tenore, F.; Etienne-Cummings, R. CPG design using inhibitory networks. *Proc. IEEE Int. Conf. Robotics and Automation*; Barcelona, Spain. Apr. 2005; p. 3682-3687.
7. Folowosele, F.; Vogelstein, R.; Etienne-Cummings, R. Spike-based MAX network for nonlinear pooling in hierarchical vision processing. *Proc. IEEE Conf. Biomedical Circuits and Systems*; Montreal, QC, Canada. Nov. 2007; p. 79-82.
8. Hodgkin A, Huxley A. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Phys*. 1952; 117(4):500–544.
9. Moris C, Lecar H. Voltage oscillations in the barnacle giant muscle fiber. *Biophys J*. 1981; 35(1): 193–213. [PubMed: 7260316]
10. Izhikevich E. Which neuron model to use? *IEEE Trans Neural Netw*. Sep. 2004 15:1036–1070.

11. Indiveri G, Linares-Barranco B, Hamilton TJ, Schaik A van, Etienne-Cummings R, Delbruck T, Liu S, Dudek P, Hifliger P, Renaud S, Schemmel J, Cauwenberghs G, Arthur J, Hynna K, Folowosele F, Saighi S, Serrano-Gotarredona T, Wijekoon J, Wang Y, Boahen K. Neuromorphic silicon neuron circuits. *Frontiers Neurosci.* 2011; 5(0)
12. Abbott LF. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res Bull.* 1999; 50(5):303–304. [PubMed: 10643408]
13. Izhikevich E. Simple model of spiking neurons. *IEEE Trans Neural Netw.* Nov.2003 14:1569–1572. [PubMed: 18244602]
14. Burkitt AN. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biol Cybern.* 2006; 95(1):1–19. [PubMed: 16622699]
15. Hille, B. *Ionic Channels of Excitable Membranes.* Sunderland, MA: Sinauer; 1992.
16. Mihalas S, Niebur E. A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Comput.* 2009; 21(3):704–718. [PubMed: 18928368]
17. Paninski L, Pillow JW, Simoncelli EP. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Comput.* 2004; 16(12):2533–2561. [PubMed: 15516273]
18. Russell AF, Orchard G, Dong Y, Mihalas S, Niebur E, Tapson J, Etienne-Cummings R. Optimization methods for spiking neurons and networks. *IEEE Trans Neural Netw.* Oct.2010 21:1950–1962. [PubMed: 20959265]
19. Dong Y, Mihalas S, Russell A, Etienne-Cummings R, Niebur E. Estimating parameters of generalized integrate-and-fire neurons from the maximum likelihood of spike trains. *Neural Comput.* 2011; 23(11):2833–2867. [PubMed: 21851282]
20. Jolivet R, Kobayashi R, Rauch A, Naud R, Shinomoto S, Gerstner W. A benchmark test for a quantitative assessment of simple neuron models. *J Neurosci Methods.* 2008; 169:417–424. [PubMed: 18160135]
21. Mihalas S, Dong Y, Niebur E. Optimal parameter search for generalized integrate-and-fire neuronal models. *Neural Computat.* 2011 submitted for publication.
22. Neftci E, Chicca E, Indiveri G, Douglas R. A systematic method for configuring VLSI networks of spiking neurons. *Neural Comput.* 2011; 23(10):2457–2497. [PubMed: 21732859]
23. Bruderle D, Petrovici M, Vogginger B, Ehrlich M, Pfeil T, Millner S, Grubl A, Wendt K, Müller E, Schwartz MO, de Oliveira D, Jeltsch S, Fieres J, Schilling M, Müller P, Breitwieser O, Petkov V, Müller L, Davison A, Krishnamurthy P, Kremkow J, Lundqvist M, Müller E, Partzsch J, Scholze S, Zuhl L, Mayr C, Destexhe A, Diesmann M, Potjans T, Lansner A, Schuffny R, Schemmel J, Meier K. A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biol Cybern.* 2011; 104:263–296. [PubMed: 21618053]
24. Saighi S, Bornat Y, Tomas J, Masson GL, Renaud S. A library of analog operators based on the Hodgkin–Huxley formalism for the design of tunable, real-time, silicon neurons. *IEEE Trans Biomed Circuits Syst.* Feb.2011 5:3–19. [PubMed: 23850974]
25. Victor J, Purpura K. Nature and precision of temporal coding in visual cortex: A metric based analysis. *J Neurophys.* 1996; 76(2):1310–1326.
26. van Rossum M. A novel spike distance. *Neural Comput.* 2001; 13(4):751–761. [PubMed: 11255567]
27. Kistler W, Gerstner W, van Hemmen J. Reduction of the Hodgkin–Huxley equations to a single-variable threshold model. *Neural Comput.* 1997; 9(5):1015–1104.
28. Russell, A.; Etienne-Cummings, R. Maximum likelihood estimation of a silicon neuron. *Proc. IEEE Int. Symp. Circuits and Systems; Rio de Janeiro, Brazil.* May 2011; p. 669–672.
29. Tenore, F.; Etienne-Cummings, R.; Lewis, MA. A programmable array of silicon neurons for the control of legged locomotion. *Proc. IEEE Int. Symp. Circuits and Systems; Vancouver, BC, Canada.* May 2004; p. 349–352.
30. Tapson J, Jin C, van Schaik A, Etienne-Cummings R. A first-order nonhomogeneous Markov model for the response of spiking neurons stimulated by small phase-continuous signals. *Neural Comput.* 2009; 21(6):1554–1588. [PubMed: 19191600]
31. Gardiner, C. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences.* Berlin, Germany: Springer-Verlag; 1985.

32. Bershadskii A, Dremencov E, Fukayama D, Yadid G. Probabilistic properties of neuron spiking time-series obtained in vivo. *Eur Phys J B, Condens Matter Complex Syst.* 2001; 24(2):409–413.
33. Beyer H, Schmidt J, Hinrichs O, Schmolke D. A statistical study of the interspike-interval distribution of cortical neurons. *Acta Biologica et Medica Germanica.* 1975; 34(3):409. [PubMed: 1189817]
34. Boyd, S.; Vandenberghe, L. *Convex Optimization.* Cambridge, U.K: Cambridge Univ. Press; 2004.
35. Levine M. The distribution of the intervals between neural impulses in the maintained discharges of retinal ganglion cells. *Biol Cybern.* 1991; 65:459–467. [PubMed: 1958731]
36. Cardoso MF, Salcedo RL, de Azevedo SF. The simplex-simulated annealing approach to continuous non-linear optimization. *Comput Chem Eng.* 1996; 20(9):1065–1080.

Biographies

Alexander Russell received the B.Sc. degree in mechatronic engineering from the University of Cape Town, Rondebosch, South Africa, in 2006, and the M.S.E degree in electrical engineering from The Johns Hopkins University, Baltimore, MD, in 2009.

Currently, he is working toward the Ph.D. degree in the Computational Sensory-Motor Systems Laboratory, The Johns Hopkins University. His research interests include optimization methods for spiking neurons and networks, mixed signal very large scale integration design, biofidelic sensory encoding algorithms, and biologically inspired algorithms for visual attention. He was a recipient of the Klaus-Jurgen Bathe Scholarship as well as the Manuel and Luby Washkansky Postgraduate Scholarship from the University of Cape Town, and the Paul V. Renoff Fellowship from The Johns Hopkins University.

Kevin Mazurek (S'10) received the B.Sc. degree in electrical engineering from Brown University, Providence, RI, in 2008, and the M.S.E. degree in electrical engineering from The Johns Hopkins University, Baltimore, MD in 2010.

Currently, he is working toward the Ph.D. degree in the Computational Motor Systems Laboratory at The Johns Hopkins University. His research interests involve modeling neural processes in mixed signal very large scale integration circuitry with emphasis on applications involving motor control.

Stefan Mihala received the B.S. degree in physics and the M.S. degree in differential geometry from West University, Timisoara, Romania, and the Ph.D. degree in physics from the California Institute of Technology, Pasadena.

Currently, he is a Postdoctoral Fellow in computational neuroscience at The Johns Hopkins University, Baltimore, MD.

Ernst Niebur received the Graduate degree and the M.S. degree (Diplom Physiker) from the Universität Dortmund, Dortmund, Germany. He received the Post-Graduate Diploma in artificial intelligence from the Swiss Federal Institute of Technology, Zurich, Switzerland, and the Ph.D. degree (Drès sciences) in physics from the Université de Lausanne, Lausanne, Switzerland. His dissertation topic was computer simulation of the nervous system of the nematode *C. elegans*.

He was a Research Fellow and a Senior Research Fellow at the California Institute of Technology, Pasadena, and an Adjunct Professor at Queensland University of Technology, Brisbane, Australia. He is currently an Associate Professor of neurosciences in the School of Medicine and of Brain and Psychological Sciences, Krieger School of Arts and Sciences, The Johns Hopkins University, Baltimore, MD. His current research interests include computational neuroscience.

Dr. Niebur was the recipient of the Seymour Cray (Switzerland) Award in Scientific Computation in 1988, the Alfred P. Sloan Fellowship in 1997, and the National Science Foundation CAREER Award in 1998.

Ralph Etienne-Cummings (S'95–M'98–SM'08) received the B.Sc. degree in physics from Lincoln University, Oxford, PA, in 1988, and the M.S.E.E. and Ph.D. degrees in electrical

engineering from the University of Pennsylvania, Philadelphia, in 1991 and 1994, respectively.

Currently, he is a Professor of electrical and computer engineering, and computer science at The Johns Hopkins University (JHU), Baltimore, MD. He is the former Director of Computer Engineering at JHU and the Institute of Neuromorphic Engineering, currently administered by the University of Maryland, College Park. He is also an Associate Director for Education and Outreach of the National Science Foundation (NSF) sponsored Engineering Research Centers on Computer Integrated Surgical Systems and Technology at JHU. His current research interests include mixed-signal very large scale integration systems, computational sensors, computer vision, neuromorphic engineering, smart structures, mobile robotics, legged locomotion, and neuroprosthetic devices.

Dr. Etienne-Cummings is the recipient of the NSF CAREER Award and the Office of Naval Research Young Investigator Program Award. In 2006, he was named a Visiting African Fellow and a Fulbright Fellowship Grantee for his sabbatical at the University of Cape Town, Rondebosch, South Africa. He was invited to be a Lecturer at the National Academies of Science Kavli Frontiers Program, held in 2007.

Fig. 1.

Adapted from [17]. The Noisy Leaky Integrate and Fire neuron model. An input current and Gaussian noise is injected into a leaky integrator. When the integrated membrane potential exceeds a threshold then the neuron emits a spike, the membrane potential is reset and a spike induced current is generated.

Fig. 2.

Recreated from [21] using the *in silico* NLIF neuron. The solid line is the First Passage Time Probability Density (Likelihood) of the NLIF firing computed with Monte Carlo simulations (30 000 iterations). The dashed line is the best fit of a normal probability distribution and the dot-dashed line is the best fit of a lognormal probability distribution. It is easily seen that the lognormal distribution approximates the likelihood much better than the normal distribution.

Fig. 3.

Raster plot showing the desired spike times (top row) against ten repeats of predicted spike times from the Maximum Likelihood parameter estimation experiment. Note the variability between predicted spike trains as a result of noise in the system.

Fig. 4.

(a) Monte Carlo simulation of membrane potential over one bursting cycle towards the end of the optimization process. Each black dot represents a recorded sample. The density of dots can be interpreted as the probability of the membrane potential assuming that value at a particular time. Note the dispersion of the probability over time due to the noise and the collapsing of probability when the neuron is reset at the beginning of each interspike interval. The numbered time intervals at the bottom of the figure denote the targeted ISIs. (b) Likelihood that the neuron spikes. The Likelihood is not smooth as a result of the small number of Monte Carlo simulations used in the optimization process. To avoid local minima the envelope of the likelihood function was used in the optimization process.

Fig. 5.

Histogram of predicted spike train interspike intervals collected over 100 s. The numbered peaks (1', 2', 3', 4') correspond to means of the predicted interspike intervals. These numbers correspond to the targeted interspike intervals (1,2,3,4) shown in Fig. 4(a). The inset shows the detail around ISI 3'.

Fig. 6.

Raster plot showing the desired spike times (top row) against ten repeats of predicted spike times from the Time Ratio Distance parameter estimation experiment. Note the variability between predicted spike trains as a result of noise in the system.