6-18-2015

System Design and Implementation of a Fast and Accurate Bio-Inspired Spiking Neural Network

Zhenzhong Wang
Florida International University, zwang009@fiu.edu

DOI: 10.25148/etd.FIDC000072
Follow this and additional works at: http://digitalcommons.fiu.edu/etd
Part of the Biomedical Commons, and the Signal Processing Commons

Recommended Citation
http://digitalcommons.fiu.edu/etd/2227

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.
SYSTEM DESIGN AND IMPLEMENTATION OF A FAST AND ACCURATE BIO-INSPIRED SPIKING NEURAL NETWORK

A dissertation submitted in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in ELECTRICAL ENGINEERING by Zhenzhong Wang

2015
To: Dean Amir Mirmiran  
College of Engineering and Computing

This dissertation, written by Zhenzhong Wang, and entitled System Design and Implementation of a Fast and Accurate Bio-Inspired Spiking Neural Network, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

_______________________________________  
Armando Barreto

_______________________________________  
Jean H. Andrian

_______________________________________  
Mercedes Cabrerozo

_______________________________________  
Wei-Chiang Lin

_______________________________________  
Malek Adjouadi, Major Professor

Date of Defense: June 18, 2015

The dissertation of Zhenzhong Wang is approved.

_______________________________________  
Dean Amir Mirmiran  
College of Engineering and Computing

_______________________________________  
Dean Lakshmi N. Reddi  
University Graduate School

Florida International University, 2015
DEDICATION

This dissertation is dedicated to the following people:

• My wife, Yanyan Pan, who has shared the life uncertainties and challenges during the completion of this dissertation, yet been always patient and supportive of my work.

• My parents, Yimin Wang and Wenhua Li, who emphasized the importance of education and always encouraged me even from thousands of miles away.

• My son, Lucas Wang, who has grown into a cute one year old boy in spite of his father spending so much time away from him working on this dissertation.
ACKNOWLEDGMENTS

I would never have been able to finish this dissertation without the guidance, support, and patience of my mentor and advisor, Professor Malek Adjouadi. I am indebted for his enthusiasm, advice, and encouragement, as well as for the financial support provided for the completion of this dissertation. I am grateful for the chance he gave me to work at the Center for Advanced Technology and Education, where I was provided with cutting-edge equipment to carry on the experiments, and where creative work is highly appreciated.

I would like to express my great appreciation to my committee members, Professor Armando Barreto, Professor Jean H. Andrian, Professor Mercedes Cabrerizo and Professor Wei-Chiang Lin, for their valuable discussions which always inspire me.

I would like to extent my gratitude to my best friends, Tan Ma and Lilin Guo, whose encouragement greatly helped me in achieving my research goals; to the FIU community and department staffs, whose professionalism is of great help to my work and life at FIU.

Finally, I would like to acknowledge the research support provided from the Department of Electrical and Computer Engineering at Florida International University, the support of National Science Foundation under grant Numbers: CNS-0959985, CNS-1042341, HRD-0833093, IIP 1338922, IIP-1230661, and CNS-1429345, and the generous support of the Ware Foundation, which facilitated the research essential to the completion of this dissertation.
ABSTRACT OF THE DISSERTATION

SYSTEM DESIGN AND IMPLEMENTATION OF A FAST AND ACCURATE BIO-INSPIRED SPIKING NEURAL NETWORK

by

Zhenzhong Wang

Florida International University, 2015
Miami, Florida

Professor Malek Adjouadi, Major Professor

Neuron models are the elementary units which determine the performance of an artificial spiking neural network (ASNN). This study introduces a new Generalized Leaky Integrate-and-Fire (GLIF) neuron model with variable leaking resistor and bias current in order to reproduce accurately the membrane voltage dynamics of a biological neuron. The accuracy of this model is ensured by adjusting its parameters to the statistical properties of the Hodgkin-Huxley model outputs; while the speed is enhanced by introducing a Generalized Exponential Moving Average method that converts the parameterized kernel functions into pre-calculated lookup tables based on an analytic solution of the dynamic equations of the GLIF model.

Spike encoding is the initial yet crucial step for any application domain of ASNN. However, current encoding methods are not suitable to process complex temporal signal. Motivated by the modulation relationship found between afferent synaptic currents in biological neurons, this study proposes a biologically plausible spike phase encoding method based on a novel spiking neuron model which could perform wavelet decomposition of the input signal, and encode the wavelet spectrum into synchronized
output spike trains. The spike delays in each synchronizing period represent the spectrum amplitudes. The encoding method was tested in encoding of human voice records for speech recognition purposes. Empirical evaluations confirm that encoded spike trains constitute a good representation of the continuous wavelet transform of the original signal.

Interictal spike (IS) is a type of transient discharge commonly found in the electroencephalography (EEG) records from epilepsy patients. The detection of IS remains an essential task for 3D source localization as well as in developing algorithms for essential in seizure prediction and guided therapy. We present in this work a new IS detection technology method using the phase encoding method with customized wavelet sensor neuron and a specially designed ASNN structure. The detection results confirm the ability of such ASNN to capture IS automatically from multichannel EEG records.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Research Purpose</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Methodology</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Structure of the Research</td>
<td>5</td>
</tr>
<tr>
<td>2. LITERATURE REVIEW</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Spiking Neuron Models</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Hodgkin-and-Huxley Model</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 Izhikevich Model</td>
<td>12</td>
</tr>
<tr>
<td>2.1.3 Integrate-and-Fire Models</td>
<td>13</td>
</tr>
<tr>
<td>2.1.4 Model Selection for ASNN Applications</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Encoding Scheme</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Rate Encoding</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Spike Time Encoding</td>
<td>17</td>
</tr>
<tr>
<td>2.3 Interictal Spike Detection</td>
<td>17</td>
</tr>
<tr>
<td>3. GENERALIZED LEAKY INTEGRATE-AND-FIRE NEURON MODEL</td>
<td>20</td>
</tr>
<tr>
<td>3.1 The Neuron Model</td>
<td>20</td>
</tr>
<tr>
<td>3.1.1 Comparison with HH models</td>
<td>22</td>
</tr>
<tr>
<td>3.1.2 Comparison with the NLIF model</td>
<td>24</td>
</tr>
<tr>
<td>3.2 GLIF Model Implementation</td>
<td>25</td>
</tr>
<tr>
<td>3.2.1 Parameters in GLIF model</td>
<td>25</td>
</tr>
<tr>
<td>3.2.2 Firing Thresholds</td>
<td>29</td>
</tr>
<tr>
<td>3.2.3 Kernel Functions</td>
<td>31</td>
</tr>
<tr>
<td>3.2.4 GLIF Model Solutions</td>
<td>35</td>
</tr>
<tr>
<td>3.2.5 Exponential Moving Average Implementation</td>
<td>37</td>
</tr>
<tr>
<td>3.3 GLIF Model Performance</td>
<td>40</td>
</tr>
<tr>
<td>3.3.1 GLIF Model Accuracy</td>
<td>40</td>
</tr>
<tr>
<td>3.3.2 GLIF Model Efficiency</td>
<td>44</td>
</tr>
<tr>
<td>3.4 Results and Discussion</td>
<td>45</td>
</tr>
<tr>
<td>4. WAVELET ENCODING METHOD FOR ASNN</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Encoding with LIF Neurons</td>
<td>51</td>
</tr>
<tr>
<td>4.2 Spike Triggered Modulation</td>
<td>54</td>
</tr>
<tr>
<td>4.2.1 Encoding Neuron Implementation</td>
<td>59</td>
</tr>
<tr>
<td>4.2.2 Encoding Human Voice Record</td>
<td>60</td>
</tr>
<tr>
<td>4.3 Results and discussion</td>
<td>61</td>
</tr>
<tr>
<td>4.3.1 Encoding Nonlinearity</td>
<td>64</td>
</tr>
<tr>
<td>4.3.2 Kernel Function Variation</td>
<td>66</td>
</tr>
<tr>
<td>4.3.3 Temporal Resolution</td>
<td>69</td>
</tr>
</tbody>
</table>
5. INTER-ICTAL SPIKE DETECTION ................................................................. 71
   5.1 Neuron Models and Network Structure ............................................... 71
   5.2 EEG Records and Inter-ictal Spike Detection...................................... 73
   5.3 Results and Discussion........................................................................ 74

6. CONCLUSIONS AND FUTURE WORK ..................................................... 76

LIST OF REFERENCES .................................................................................. 79

APPENDICES ................................................................................................. 88

VITA ............................................................................................................... 91
<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1 Circuit diagram of HH neuron model</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.1 A Pair of Synapse and Neuron. Input spikes generated by Poisson Process with random connection weights stimulate the neuron and induce regular spikes of action potentials</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3.2 The firing threshold of the membrane voltage. (a) Probability distribution of the local maximums of membrane voltage. The left region of this distribution indicates those peaks did not induced action potentials; the right region of this distribution shows the action potential peaks. Almost no local maximum could be found in between; (b) Cumulative distribution of the regular peaks; the firing threshold uth is defined so that 95% of regular peaks happens below it.</td>
<td>30</td>
</tr>
<tr>
<td>Figure 3.3 Mean value of Membrane voltage (black squares using left and bottom axes) and conductance of sodium and potassium channel (red rounds and green triangles using right and top axes) versus survival time. The standard variances are indicated by the error bars at all sample points individually.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3.4 GEMA window width according to the survival time. The simulation step size is 0.1 ms</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.5 F1 scores at different firing thresholds and absolute refractory period lengths. (a) F1 scores for NLIF model; (b) F1 scores for GLIF model</td>
<td>44</td>
</tr>
<tr>
<td>Figure 3.6 Comparison of membrane voltage evolution under current input using a Poisson input spike train for NLIF, GLIF and HH models. Vertical bars marked under the membrane voltage of NLIF model indicate its spiking locations.</td>
<td>46</td>
</tr>
<tr>
<td>Figure 3.7 Zoom view of simulated membrane potentials in GLIF, HH and NLIF models</td>
<td>48</td>
</tr>
<tr>
<td>Figure 3.8 Missed Fire Rates and Accidental Fire Rates of NLIF and GLIF model under various simulation step size</td>
<td>50</td>
</tr>
<tr>
<td>Figure 4.1 Structure of the Two-Stage Modulate-and-Integrate Module</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.2 Time course of variables in one WSN with $\sigma = 5.64$ ms. Red vertical dash lines indicates the arrival times of spikes in Cint; green dash lines indicates the arrival times of spikes in Cenc.</td>
<td>62</td>
</tr>
<tr>
<td>Figure 4.3 Comparison of WSN encoding with Continuous Wavelet Transform at corresponding translations. Green lines bars output spikes from the WSN array</td>
<td>63</td>
</tr>
</tbody>
</table>
Figure 4.4 Logarithm relationship of the input intensity and output spike delay: (a) the relationship of positive threshold WSNs; (b) the relationship of negative threshold WSNs.

Figure 4.5 Comparison between WSN encoding using Mexican-hat wavelet and Alpha function. The $\sigma_m$ are the time scales used for WSNs with Mexican-hat wavelet kernels while $\sigma_a$ are the corresponding time scales used for WSN with Alpha function kernels.

Figure 5.1 Network structure to detect interictal spikes from EEG records.

Figure 5.2 Encoding of interictal spikes in a segment of multi-channel EEG recordings: (a) the waterfall plot of the recorded EEG potentials; (b) the pseudo color mapping of the encoded output spike delays; (c) detailed view of the IS wave found at approximately 18th second in the upper portion, and output spikes generated by WSNs in the lower portion.
## ABBREVIATIONS AND ACRONYMS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AER</td>
<td>Address-Event Representation</td>
</tr>
<tr>
<td>AFR</td>
<td>Accidental Fire Rate</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ASNN</td>
<td>Artificial Spiking Neural Network</td>
</tr>
<tr>
<td>BSA</td>
<td>Bens Spiker Algorithm</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>EMA</td>
<td>Exponential Moving Average</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>GEMA</td>
<td>Generalized Exponential Moving Average</td>
</tr>
<tr>
<td>GLIF</td>
<td>Generalized Leaky Integrate-and-Fire</td>
</tr>
<tr>
<td>GPGPU</td>
<td>General Purpose Graphic Process Unit</td>
</tr>
<tr>
<td>HH</td>
<td>Hodgkin and Huxley</td>
</tr>
<tr>
<td>IF</td>
<td>Integrate-and-Fire</td>
</tr>
<tr>
<td>IS</td>
<td>Interictal-Spike</td>
</tr>
<tr>
<td>LIF</td>
<td>Leaky Integrate-and-Fire</td>
</tr>
<tr>
<td>LTP</td>
<td>Long-Term Potentiation</td>
</tr>
<tr>
<td>MAT</td>
<td>Multi-time-scale Adaptive Threshold</td>
</tr>
<tr>
<td>MFR</td>
<td>Missed Fire Rate</td>
</tr>
<tr>
<td>NEST</td>
<td>NEural Simulation Tool</td>
</tr>
<tr>
<td>NLIF</td>
<td>Normal Leaky Integrate-and-Fire</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equation</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>PE</td>
<td>Phase Encoding</td>
</tr>
<tr>
<td>PSC</td>
<td>Post-Synaptic Current</td>
</tr>
<tr>
<td>RK4</td>
<td>Forth-order Runge-Kutta</td>
</tr>
<tr>
<td>SNN</td>
<td>Spiking Neural Network</td>
</tr>
<tr>
<td>STDP</td>
<td>Spiking-Timing-Dependent Plasticity</td>
</tr>
<tr>
<td>WSN</td>
<td>Wavelet Sensor Neuron</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1 Background

Artificial Neural Network (ANN) is one important class of machine learning methods which are inspired by the features of biological neurons, and the organization structures of biological nervous systems. Classified by the computing unit dubbed as the neuron model, ANN has evolved for three generations [1]. In the first generation of ANN, the neuron model simply sums all inputs, compares the summation with a threshold value, and generate an output ‘1’ if the summation overshoots the threshold, and ‘0’ otherwise. Such a hard threshold mechanism incorporated in the first generation of ANN makes it inadequate to manipulate analog inputs, subsequently substituted by smoother activation functions such as R function, sigmoid function or arc tangent function [2] in the second generation of ANN. Both the first and second generation of ANN could be treated as rate encoding simplification of a biological nervous system, i.e., the input and output values to/from ANN neurons could be considered as the average firing rate of a biological neuron in a short time window. However, later neuroscience research shows that, biological neurons might convey and manipulate information using the timing of individual fires [3], which leads to the development of the third generation of ANN, the Spiking Artificial Neural Network (ASNN).

An ASNN is constructed by connecting a multitude of spiking neurons. Each spiking neuron is an electrical device or mathematical model that accumulates temporal input
signals, simulates the neuroelectrical behaviors as found in biological neurons, and generate an output pulse under proper conditions. For example, one simplest spiking neuron model could integrate input values into a state variable $u$. The neuron is termed “fired” and generating an output pulse called “spike” when its state variable $u$ is greater than a threshold, and the state variable is reset to a level lower than the threshold in preparation for the next fire. Such neuron model is called the “Integrate-and-Fire” (IF) model. There are other models that incorporate more complicated behaviors such as intrinsic firing, post-fire inhibition, bursting firing, etc. The development of neuron models are mostly urged by either the demand of accuracy in reproducing the biological neuron behaviors, or by the requirement of efficiency in the simulation of large-scale networks.

In ASNN, a pair of neurons are directionally connected by a “synapse,” which is an abstraction of the biological synapse. Each synapse in ASNN stores a state variable called its weight, which represents the connection strength between the two neurons it connects. When the synapse receives an incoming spike from its pre-synaptic neuron, it will generate a special waveform related to the synapse weight to stimulate its post-synaptic neuron. The on-set of such waveform is usually postponed for a certain period after the receiving of an incoming spike, such period is called the synaptic delay.

The learning of ASNN indicates the rules to modify the synapse weights, and consequently changes the firing pattern of output neurons. There are several learning methods proposed for ASNN, such as spike-timing-dependent plasticity (STDP), which
uses the firing time difference between the pre-synaptic and post-synaptic neurons to modify the synapse weight [4]; distal reward reinforcement which considers the global training error for STDP [5]; rank order coding which rewards synapses that provide earlier output waves to one neuron, and punishes synapses that generate late waves [6]; and methods derived from traditional ANN training practices [7]–[10]. These methods are well-developed to train the ASNN in either a supervised way or an unsupervised manner.

ASNN is considered more biologically plausible in the sense that its neuron model mimics the neuroelectrical behavior found in biological neurons, and the connections between neurons in an ASNN are similar to their biological counterparts. Such biologically plausible construction ensures ASNN a greater processing power in manipulating temporal signals, and better robustness in complex patterns learning [11].

1.2 Research Purpose

This dissertation focuses on the implementation of biologically plausible ASNN, and its applications on biomedical signals. The goal of this work is threefold:

1. Developing a novel biologically plausible neuron model which is both accurate and efficient in mimicking neuroelectrical dynamics for ASNN. The model should have the following features:

   (i) Similar behavior to that of a well-known accurate model.
(ii) Clear model definition and parameter extraction procedure.

(iii) Competitive efficiency compared to other simple neuron models.

2. Developing an encoding method to convert temporal signals into meaningful spike trains ready to feed into ASNN. The encoding method should be easy to implement on common ASNN simulation platforms, and the converted spike train should convey the spike time encoded information related to the important features in the input signal.

3. Applying ASNN to biomedical signal processing. Specifically in this work, we would like to use the spike time encoding method to extract important features from electroencephalogram (EEG) records that would enhance the prospects for analysis and diagnosis for patients with epilepsy.

1.3 Methodology

In this work, we applied analytical and computational methods for the following tasks:

1. Modeling the spiking neuron using ordinary differential equations (ODE). The behavior of the new neuron model is analyzed by solving the model ODE analytically. A novel fast implementation method for this new model has also been developed by applying functional transformations to the solution of model ODE. Numeric fitting methods were adopted to extract model parameters from experiments on an existing accurate neuron model with purposely designed random inputs. The new neuron model was implemented numerically both in the Matlab environment and on a
general ASNN simulation platform, with accuracy and efficiency compared to existing neuron models using iterative numerical analysis methods and probability theory.

2. We expanded the structure of a simple spiking neuron model with the idea that multiplication instead of summation of the inputs to one neuron could perform the transformation similar as wavelet decomposition to the inputs. The proof of concept was done by designing such special structured neuron and solving its ODE analytically. The encoding method was implemented on a general ASNN simulation platform by designing and integrating a new neuron model for this platform.

3. We extracted features from EEG records which could help seizure detection by simulating a specially designed ASNN structure with customized neuron models. The entire simulation was done in the Python interface of the ASNN simulation platform.

1.4 Structure of the Research

This dissertation introduces novel methods for implementing ASNN and applying it to biomedical signal processing. The dissertation is structured into six chapters, starting from the current chapter that outlines the research background and purpose.

Chapter 2 introduces the current development in the implementation and application of ASNN. Multiple existing neuron models are reviewed in this chapter, and the selection criterion for a good neuron model is discussed. Existing rate encoding and spike time
encoding methods are also reviewed in this chapter to provide a retrospective on ASNN encoding schemes. Literature related to the inter-ictal spike detection are summarized at the end of this chapter.

Chapter 3 proposes a design model and implementation of a new Generalized Leaky Integrate-and-Fire (GLIF) by innovative modifications of the Normal Leaky Integrate-and-Fire (NLIF) model, so that comparable accuracy of the Hodgkin and Huxley (HH) model is attained, while the computational complexity of the model is significantly reduced. This GLIF model could be solved analytically and a novel numerical method is proposed to simulate the neuron model, so that lookup tables could be built before the simulation process.

Chapter 4 introduces a preprocessing unit for the Leaky Integrate-and-Fire (LIF) spiking neurons. The assumption is that a neuron model combining the preprocessing unit with a LIF neuron could be used to encode analog signals with a wide frequency range. We demonstrate in this chapter that our preprocessing unit could decompose the input signal into the wavelet spectrum, and further encode the spectrum amplitude into the delay amount between output spikes and the clock signals. Empirical results of Phase Encoding (PE) of speech signals are provided, with linearity, temporal resolution issues and possible extension of the encoding method discussed.
Chapter 5 elaborates on the neuron models and network structure we designed to detect interictal spikes (IS) in EEG records. The ASNN parameters required for this detection task are described in this chapter, followed by the detection results and discussions.

Chapter 6 summarizes the findings in this dissertation, and provides possible directions for related research in the future.
2. LITERATURE REVIEW

This chapter reviews the literature on spiking neuron models presenting existing encoding methods in determining input signals for ASNN, and methods developed for the purpose of detecting interictal spikes in EEG recordings.

2.1 Spiking Neuron Models

The computing unit in an ASNN is usually a mathematical model or electronic unit, which characterizes the membrane potential dynamics of a neuron cell. We will introduce several of most popular neuron models in this section. These neuron models vary from the most complicated ones fitted to mimic real biological neurons, to the simplest ones, which only abstract the most important electrophysiology features. However, the challenge remains in selecting a proper model for the ASNN because of the difficulty in balancing accuracy and complexity of the mathematical model while attempting to reproduce the dynamic behavior of a neuron model.
2.1.1 Hodgkin-and-Huxley Model

As shown in Figure 2.1, there exist some specific proteins called “gates” inside the cell membrane, which could transport ions from one side of the membrane to the other side. Some of the gates are called ion channel, which only passively allow specific types of...
ions to permeate from the higher ionic concentration side to the lower side. There are also some gates called ion pumps due to their ability of actively transporting ions through the cell membrane despite the ionic concentrations. When the pumping and diffusing of a certain ion achieve a dynamic equilibrium, the electrical potential difference between the cell membrane is called the reversal potential.

Hodgkin and Huxley studied two major types of ion gates related to the sodium ions and potassium ions on the giant axon neuron of a squid [12]–[16], due to which the electrophysiology properties of this neuron could be represented by the circuit shown in Figure 2.1. The membrane is considered as a capacitor, which could be charged and discharged by injected currents $I_{\text{all}}$, and ionic currents from inside the cell. The power sources $E_K$ and $E_{Na}$ are the reversal potentials of ion K and Na, $g_K$ and $g_{Na}$ are the expressed conductances of the K ion channel and the Na ion channel, respectively, both of which vary with the membrane potential $u_m$. All other ion channels in this neuron are considered to be unrelated to the membrane potential, and thus represented by a constant conductance $g_L$. The reversal potential for ions with constant conductance channels are considered as a lumped potential $E_L$. The dynamic of membrane potential is governed by the following ODE:

$$C_m \frac{du_m}{dt} = I_{\text{all}}(t) - g_{Na}(m,n)(u_m - E_{Na}) - g_K(h)(u_m - E_K) - g_L(u_m - E_{Na})$$

The conductances $g_{Na}$ and $g_K$ are functions of gating parameters $m$, $n$, and $h$, which are related to the portions of opening gates in a neuron cell for the fast-acting sodium channel,
slow-acting sodium channel and potassium channel. Hodgkin and Huxley designed the well-known voltage clamp experiment on the giant axon neuron to analyze the dynamics of these gating parameters, and fitted the functions $g_{Na}$ and $g_{K}$ to polynomial functions:

$$\frac{dm}{dt} = \alpha_m(u_m)(1 - m) - \beta_m(u_m)m$$

$$\frac{dn}{dt} = \alpha_n(u_m)(1 - n) - \beta_n(u_m)n$$

$$\frac{dh}{dt} = \alpha_h(u_m)(1 - h) - \beta_h(u_m)h$$

$$g_{Na} = \bar{g}_{Na}m^3h$$

$$g_{K}(n) = \bar{g}_{K}n^4$$

Where $\bar{g}_{Na}$ and $\bar{g}_{K}$ are the constants of maximum conductance for sodium and potassium channels. Parameters $\alpha_m$, $\alpha_n$, and $\alpha_h$ are the changing speeds of gates related to $m$, $n$, and $h$ from open state to close state, while $\beta_m$, $\beta_n$, and $\beta_h$ are the changing speed in the opposite direction. All these changing speeds are unitless univariate functions that depend solely on $u_m$, with outcome ranges between zero and one. The functions fitted by Hodgkin and Huxley are:

$$\alpha_m = \frac{2.5 - 0.1u_m}{\exp(2.5 - 0.1u_m) - 1}$$

$$\beta_m = 4\exp(-u_m / 18)$$

$$\alpha_n = \frac{0.1 - 0.01u_m}{\exp(1 - 0.1u_m) - 1}$$

$$\beta_n = 0.125\exp(-u_m / 80)$$

$$\alpha_h = 0.07\exp(-u_m / 20)$$

$$\beta_h = \frac{1}{\exp(3 - 0.1u_m) + 1}$$

Equations (2.1), (2.2), and (2.3) constitute the original HH model. Since the HH model could accurately describe the electrophysiology dynamics of a neuron cell, and all
parameters have clear physical meanings, many researchers tried to expand the HH model to other types of neurons with more complicated ion channels [17]–[22].

Pospischil et al. [23] reviewed these ion channels and proposed a general form of HH model for neurons as postulated in the human brain. Although the HH model has been proven to be accurate in reproducing the spiking activities of a biological neuron, its applications in ASNN are still rare, due to its computational complexity.

2.1.2 Izhikevich Model

By applying bifurcation methodologies to the HH model, Izhikevich proposed a simpler two variable model [24], [25]. The ODE for the Izhikevich model is defined by

$$\frac{du_m}{dt} = a(bv-u_m)$$
$$\frac{dv}{dt} = 0.04v^3 + 5v + 140 - u + I_{\text{ill}}$$

with post-fire resetting:

$$\text{if } v \geq 30 \text{ mV, then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$$

By adjusting the four variables $a$, $b$, $c$, and $d$, the Izhikevich model could reproduce a couple of different spiking patterns found in the biological neuron, such as the regular spiking, intrinsically bursting, fast spiking, among others.
Although the conductance and current changes of the ion-channels are not fully described, the Izhikevich model, which is expressed in a concise form, remains successful in reproducing different types of neuronal dynamics. This was widely acknowledged by researchers working on large-scale neural network practices [26], [27].

2.1.3 Integrate-and-Fire Models

Despite the neuron models abstracted from the detailed biophysical characters, there are also phenomenal models which concentrate on the overall behaviors of the membrane voltage. These models are simpler than the detailed neuron models and are thus preferred in ASNN implementations [28]. One of the simplest phenomenal model is the IF model, which treats the neuron membrane as a capacitor, which accumulates the input currents. The threshold is a level of membrane voltage above which an output spike will be fired and the membrane voltage will be dropped to a lower reset value.

Considering that IF model is too coarse to capture any complicated biological neuron features, a variety of refinements have been made to improve its performance. One of the most commonly used variations is the LIF model [29], with a linear leaky resistor inserted in the membrane capacitor. The LIF model is a valuable tool for analyzing key neuron properties because of the way it models one of the key features of a neuron’s membrane: the membrane’s conductance. Also, the model ODE can be solved analytically, thus providing intuitive insight to the biological neuron activities [30].
The LIF model has been further improved by introducing other biologically plausible features, such as nonlinear leakage term [31] and moving thresholds [32], [33]. The variability of thresholds in the moving threshold models equipped the LIF model with a refractory period, which is argued to be very important in the cognition process of spiking neural network (SNN) [29], [34]. Different ways that regulate the changing of thresholds have been proposed [35], [36]. The multi timescale adaptive threshold (MAT) model proposed by Kobayashi et al. [37] is preeminent in its accuracy of reproducing the neuron behaviors, leading the competition of neuronal activity challenge launched by the International Neuroinformatics Coordinating Facility in 2009 [38].

2.1.4 Model Selection for ASNN Applications

Both detailed biophysical neuron models and simplified phenomenal neuron models have been considered in ASNN implementations, yet ASNN applications vary according to the type of neuron models being used. The HH model and the Izhikevich model have been successfully used in simulating functional blocks of a biological nervous systems [39], [40] due to their ability to simulate complicated single neuron activities. Yet these applications require large-scale ASNN implementation, making phenomenal models the preferred ones for to their simplicity in structure and efficiency in their simulation process [28]. The LIF model with its plausible biological features has been proven to work well in biological SNN behavior analysis and computer-aided recognition and classification tasks [32], [41]–[43]. However, when the computational requirements are substantial, as in the case of large-scale ASNN implementation, the neuron model becomes inadequate, and as such, the simplest LIF models or even IF models will be
preferable [44], [45], [10], [46]–[48]. Ros et al. [49] suggested in their ASNN implementation that network simulation could be optimized when lookup tables can be pre-calculated for the neuron model before the simulation starts.

2.2 Encoding Scheme

The most significant difference between ASNN and traditional neural networks is that information in ASNN is represented by spike trains which are a series of pulses with timings of interests. There are mainly two kinds of interpretations developed in signal processing applications about how information is related to spike trains: (i) the rate encoding, which assumes that the information is encoded by the counts of spikes in a short time window; and (ii) the spike time encoding which considers information carried at the exact time of each pulse in the spike train. Although the mechanisms for data representation and analysis using biologically-inspired neural networks is still under development, empirical evidence has shown that spike time encoding might be more reliable in explaining experiments on the biology of nervous systems [50], [51]. Both rate encoding and spike time encoding essential in ASNN applications.

2.2.1 Rate Encoding

The easiest way to rate encode an analog signal is to feed it to a Poisson neuron, which fires output spikes at probability proportional to its membrane potential, thus making its firing rate within a short time window proportional to the amplitude of the input signal. Such an encoding method has been adopted by Sprekeler et al. [52]. and Keer et al. [53]
in order to analyze the recurrent ASNN behaviors. Although Poisson neuron model is simple and suitable for theoretical analysis, it was rarely implemented in real-world applications due to its inaccuracy in mapping analog signals to spike trains. De Garis et al. [54] introduced another rate encoding method which deconvolves the input signal into its individual spike responses, so that the post-synaptic potential of the encoded spike train could be quite similar to the original signal. Schrauwen and Van Campenhout [55] improved algorithm proposed by De Garis et al. by optimizing the deconvolution threshold yielding the so-called Bens Spiker Algorithm (BSA). BSA has been used widely as a rate encoding method for ASNN applications [56]–[58]. The major problem of this type of rate encoding is that an averaging time window is required for each sampling of the input signal, which as a consequence limits the temporal resolution of the encoded signals.

In order to overcome this drawback, receptive fields are introduced by other researchers to improve the temporal resolution [47], [59], where input signals are first decomposed by Gaussian windows with variant shifts of the window center, and then fed to an array of neurons which convert them to multiple spike trains. Address-Event Representation (AER) is an asynchronous protocol designed for analog neural system simulation platforms [60]. However, AER is also referred to as an encoding method by other groups of researchers [61]–[63]. When used as an encoding method, encounters of “ON” and “OFF” events in the input signals are registered by AER to generate corresponding output spikes. The “ON” and “OFF” events in AER indicate the time when a change in the input signal either exceeds a positive threshold or fall behind a negative threshold. Under such
definition, AER could be treated as a rate encoding method with regards to the derivatives of the input signal.

### 2.2.2 Spike Time Encoding

Synchronized spike time encoding, dubbed as Phase Encoding (PE), was also widely used in ASNN application. A simple implementation of PE could be realized by linearly mapping the input signal to the delay of spikes within each synchronizing period [34]. This implementation of PE requires the input signal either to be static or vary at frequencies much lower than the synchronizing frequency. Temporal receptive fields could also be utilized for PE to improve the encoding resolution [64], [65]. To be more biologically plausible, Rumbell et al. [66] introduced a synchronizing method which considered spiking neurons as PE units instead of performing linear mapping between analog values and spike delays. Receptive fields in this study were applied to the amplitude dimension instead of the temporal dimension, which yielded good performance for static input data. However, PE method which could accurately encode temporal signals is still under development.

### 2.3 Interictal Spike Detection

EEG signals are measurements of brain neurophysiology activities, and thus serve as a fundamental way to diagnose many neurological disorders [67], among which diagnosis and prediction of seizures for patients with epilepsy is one important application of EEG signal analysis [68]. After analyzing EEG signals from epilepsy patients and comparing
them with those from controls, researchers found that there is a special kind of transient EEG discharges, dubbed as inter-ictal spikes (IS), which occur frequently in the patients’ EEG recordings, and is highly correlated with epilepsy [69]. Although the relationship between IS and epileptic seizures are not fully understood so far, neurologists believe that IS could be initiation precursor to seizures [70], or the causation of seizures in the way that they could be sufficient to induce long-term potentiation (LTP) of synapse between neurons and cause excessive network synchronization [71]. In either possibility mentioned above, recognition of IS from EEG signals could help in the diagnosis and prediction of epileptic seizures.

The recognition of IS from EEG recordings depends on the characters of IS, including the temporal shape, frequency features, and the synchronization and causation among multiple recording channels. Early attempts for automatic detection of IS were based on extracting peaks with certain amplitude, duration, and sharpness [72], yet such methods are not robust to learn the difference of IS shapes among different patients, nor able to suppress variance introduced by different measurement devices or environment noise. In order to deal with the non-stationary nature of the shape of IS, Latka and Was [73] introduced wavelet decomposition method to analyze EEG signals and detect IS. Latka and Was adopted Mexican hat wavelet in the EEG signal decomposition, and applied spike detection on the resulting wavelet coefficients. Adjouadi et al. [74] applied the discrete Walsh transform instead of continuous wavelet in the EEG signal decomposition, and designed spike duration filter mechanism together with an adaptive threshold to further increase the detection accuracy. ANN is another promising and preferred tool to
detect IS. Researchers are interested in utilizing the learning power provided by ANN to memorize the IS specific shapes or characterizing features, and use the trained ANN to detect IS from new EEG records [75]–[78]. Since the neural network evolved to the third generation, the processing power of ASNN in manipulating temporal signals inspired many experiments which used ASNN to analysis raw EEG data[34], [56], [79]. However, algorithms using ASNN to detect special transients, such as IS, from EEG records are still rare.
3. GENERALIZED LEAKY INTEGRATE-AND-FIRE NEURON MODEL

3.1 The Neuron Model

The following equations define the newly developed spiking neuron model, which is a modification of the NLIF model. These modifications are intended to resolve the contentious issues of reducing significantly the computational complexity of the HH model while maintaining the high accuracy in reproducing effectively the dynamic behavior of a neuron cell’s membrane voltage. These mathematical derivations consolidate both of these models (NLIF and HH) by introducing the variable membrane conductance and bias current to the LIF model in order to reach the high accuracy of the HH model. A Generalized Exponential Moving Average (GEMA) is then used as means to reduce the processing time.

These next two equations define the dynamics of the membrane voltage $u$ of a neuron with membrane capacitance $C_m$:

\[
C_m \frac{du(t)}{dt} = i_s(t) - u(t) \cdot g_{\text{kern}}(t-t^f) + i_{\text{kern}}(t-t^f),
\]

(3.1)

where $t^f$ is the most recent firing time:

\[
t^f : u(t^f) = u_{\text{th}} \text{ and } \left. \frac{du}{dt} \right|_{t^f} > 0.
\]

(3.2)
Function $g_{\text{kern}}$ in (3.1) is a temporal function of conductance which describes the changing in leaky resistor after each output fires; while function $i_{\text{kern}}$ describes the bias current of the neuron which could stimulate an action potential. The “kern” subscript stand for “kernel” functions, indicating that these two functions are repeated each time the neuron fires.

Stimulation of this model is expressed as a summation of currents from all synapses connected to this neuron, denoted by $i_s$ in (3.1). The firing time of this neuron defined by (3.2) is updated when the membrane voltage crosses a threshold $u_{\text{th}}$. Immediately after each output fire, a new $t_f$ is used to calculate $g_{\text{kern}}$ and $i_{\text{kern}}$. The time interval $t - t_f$ is referred to as the survival time, i.e., the length in which the neuron stays quiescent since the last time it fired.

We assume that the kernel functions could be described through summation terms of a set of functions $f_j$ as follows:

$$
\begin{align*}
  g_{\text{kern}}(t) &= k_L + \sum_j f_j(t) \\
  i_{\text{kern}}(t) &= i_L + \sum_j W_j f_j(t),
\end{align*}
$$

(3.3)

where $k_L$ and $i_L$ are the constant leaking conductance and bias current when the neuron is quiet, $f_j$ are bell-shaped functions which could fit the trajectories of conductance for all ion-channels during an action potential, and $W_j$ are the resting potentials for those channels. The bell-shaped curves $f_j$ as used here are the derivatives of generalized sigmoid functions $\Phi_j$, which are formulated as follows:
\[ \Phi_j(t) = \frac{A_j}{1 + e^{-(\tau_j \mu_j) / \nu_j}} \]  

(3.4)

\[ f_j(t) = \frac{d\Phi_j}{dt} = \frac{A_j \nu_j}{\mu_j} = \frac{e^{-(\tau_j \mu_j) / \nu_j}}{1 + e^{-(\tau_j \mu_j) / \nu_j}} \]  

(3.5)

where \( A_j, \mu_j \) and \( \nu_j \) are empirical parameters controlling the bell amplitude, shape and location, which need to be fitted to the experimental ionic current data.

### 3.1.1 Comparison with HH models

Since the HH model is ideal for reproducing electrophysiological responses accurately for any types of biological neurons, it is used as the reference model in this study for the extraction of GLIF model parameters and estimation of GLIF model accuracy. The generalized HH model suggested by Pospischil [23] could be formulated as:

\[ C_m \frac{du}{dt} = i_s(t) - i_{ion}(u,t) \]  

(3.6)

\[ i_{ion}(u,t) = \sum_j \bar{g}_j m_j^{n_j} n_j^{q_j} (u - E_j), \]  

(3.7)

where \( i_{ion} \) is a summation over all possible ionic currents, with \( \bar{g}_j \) defining the maximum conductance, and \( E_j \) being the resting potential. The \( m_j \) and \( n_j \) parameters define the gating for each ion channel, with \( p_j \) and \( q_j \) being their respective power indices. Parameters \( m_j \) and \( n_j \) evolve according to the following dynamic equations:

\[ \frac{dm_j}{dt} = \alpha_m(u)(1 - m_j) - \beta_m(u)m_j \]  

\[ \frac{dn_j}{dt} = \alpha_n(u)(1 - n_j) - \beta_n(u)n_j, \]  

(3.8)
where $\alpha_{m,j}$, $\beta_{m,j}$, $\alpha_{n,j}$, and $\beta_{n,j}$ are constants or explicit functions only related to the membrane voltage $u^1$. It should be noted that, the contribution of the constant leaking channel in the ionic current, which is usually expressed as an individual term, could also be expressed in the summation of (3.7) by simply defining $m_j(0) = 1$, $n_j(0) = 1$, $\alpha_{m,j} \equiv 0$, $\beta_{m,j} \equiv 0$, $\alpha_{n,j} \equiv 0$ and $\beta_{n,j} \equiv 0$ for the leaking channel $j$.

In order to compare (3.6) with (3.1), the following relations are assumed:

\[
\begin{align*}
g^*(t) &= \sum_j \bar{g}_j m_j^n n_j^n \\
\quad i^*(t) &= \sum_j \bar{g}_j m_j^n n_j^n E_j.
\end{align*}
\]

(3.9)

Functions $g^*$ and $i^*$ are equivalent to $g_{\text{kern}}$ and $i_{\text{kern}}$ when the following conditions are satisfied:

(i) Functions $g^*$ and $i^*$ repeat the same trajectory after each time the neuron fires;

(ii) Functions $g^*(t - t^f)$ and $i^*(t - t^f)$ could be fitted well by (3.3), with $t^f$ being the most recent firing time of the neuron.

The next section elaborates on the supposition that if the stimulation is a train of random spikes generated by a Poisson Process, the HH model could then be approximated by the GLIF model.

---

1 Functions $\alpha_{m,j}$, $\beta_{m,j}$, $\alpha_{n,j}$, and $\beta_{n,j}$ are explicit functions of $u$ fitted to the data from voltage clamp experiments. Although these functions should always yield dimension $s^{-1}$ to balance the equation, the forms of these functions proposed by Hodgkin and Huxley in their original model ODEs and most of their derives incurred unit mismatches. Such unit mismatch has never been brought up because the model is accurate as long as mV unit is used for the membrane voltage, which was the default configuration in most biological experiments. Since the unit for all voltages have been set to mV in this paper, functions $\alpha_{m,j}$, $\beta_{m,j}$, $\alpha_{n,j}$, and $\beta_{n,j}$ are kept in their original form as in Hodgkin and Huxley’s works.
3.1.2 Comparison with the NLIF model

A NLIF model having a linear leaky resistor could be described by the following equation [80]:

\[ \tau_m \frac{du}{dt} = Ri - u, \]

(3.10)

where \( R \) is the resistant of a linear resistor and \( \tau_m \) is a time constant for the integrator. A NLIF neuron has no explicit action potential but only the firing time defined in the same way as in the GLIF neuron in (3.2). The membrane voltage is reset to \( u_{\text{rest}} < u_{\text{th}} \) right after the neuron fires.

As \( t \) in (3.3) is significantly increased, both \( g_{\text{kern}} \) and \( i_{\text{kern}} \) will tend to constant values:

\[ g_{\infty} = \lim_{t \to +\infty} g_{\text{kern}}(t) = k_L, \]
\[ i_{\infty} = \lim_{t \to +\infty} i_{\text{kern}}(t) = i_L. \]

(3.11)

If the bias \( i_L \) is set to zero in (3.3), the NLIF model could be considered as a degraded form of the GLIF model for long survival times, and where the time constant and linear resistant can be defined as follows:

\[ \tau_m = C_m / g_{\infty}, \]
\[ R = 1 / g_{\infty}. \]

(3.12)

In other words, the NLIF model describes the membrane potential when the neuron stays quiescent, while the GLIF model modified the NLIF model by introducing the variable membrane conductance and bias current, yielding a better accuracy in approximating the HH model in the post-fire region. The next section provides the derivations and empirical
evaluations that led to these modifications and describes the methods used for establishing and implementing the GLIF model.

3.2 GLIF Model Implementation

This section describes the modifications that were made on the NLIF and the consequential reasons for (3.1) introducing the variable membrane conductance and bias current to make it accurate, and (3.2) making use of the GEMA approximation of the GLIF model solution in order to optimize the computational requirements and speed up its simulation process.

3.2.1 Parameters in GLIF model

There are several parameters that need to be determined empirically in (3.1) and (3.2) for generating the GLIF model.
Although in the most realistic simulation of SNN, each neuron receives input from multiple synapses with behaviors governed by synaptic dynamics and system learning mechanism [4], Zhen and Leung [81] identified that current based synapse models could serve as good approximations of the synapse dynamic when the interspike intervals of the spike train fed to the synapses are uniform. In this paper, we used one current based synapse stimulated by Poisson distributed spike train as the overall current stimulation to the neuron, as shown in Figure 3.1. However, the synapse weight was assumed to be a

![Figure 3.1 A Pair of Synapse and Neuron. Input spikes generated by Poisson Process with random connection weights stimulate the neuron and induce regular spikes of action potentials.](image)

spike train fed to the synapses are uniform. In this paper, we used one current based synapse stimulated by Poisson distributed spike train as the overall current stimulation to the neuron, as shown in Figure 3.1. However, the synapse weight was assumed to be a
random variable with uniform distribution which value changes for each spike in the input spike train: suppose the input spike train has \( m \) spikes arriving at time instants \( s^{(1)} \), \( s^{(2)} \), ..., \( s^{(m)} \), the connection weights will be \( w_1, w_2, ..., w_m \), individually.

The spike train could be converted into synaptic current by an amplitude modulated spike-shaped current source model as follows:

\[
i_s(t) = \sum_j w_j i_s^*(t - s^{(j)}) \Theta(t - s^{(j)})
\]

\[
i_s^*(\tau) = I_{s,\text{max}} \frac{\tau}{\tau_s} e^{-\tau/\tau_s},
\]

(3.13)

where \( i_s^* \) is the shape function for a single input current spike, with maximum current \( I_{s,\text{max}} = 23 \mu A/cm \) and rising/decay time \( \tau_s = 2 \text{ ms} \), and with \( \Theta \) denoting a Heaviside step function.

The HH model as suggested by Hodgkin and Huxley [15] is used here with two active ion channels for demonstration purposes. The currents in the sodium channel, the potassium channel and the linear leaky channel are as formulated below:

\[
i_{\text{ion}} = g_{\text{Na}} (u - E_{\text{Na}}) + g_{\text{K}} (u - E_{\text{K}})
+ g_{\text{L}} (u - E_{\text{L}})
\]

(3.14)

\[
g_{\text{Na}} = \bar{g}_{\text{Na}} m^3 h
\]

\[
g_{\text{K}} = \bar{g}_{\text{K}} n^4.
\]

The maximum conductance for sodium, potassium and leaking channels are found to be \( \bar{g}_{\text{Na}} = 120 \text{ mS/cm}^2 \), \( \bar{g}_{\text{K}} = 36 \text{ mS/cm}^2 \), and \( \bar{g}_{\text{L}} = 0.3 \text{ mS/cm}^2 \), respectively. The
corresponding reversal potentials adopted were $E_{Na} = 115 \text{ mV}$, $E_{K} = -12 \text{ mV}$, and $E_{L} = 10.6 \text{ mV}$.

An experiment to determine parameters in the GLIF model was designed using the following steps:

(i) Choose randomly a frequency $\lambda_{P}$ between 1 Hz and 1000 Hz.

(ii) Generate a spike train lasts 1000 ms by a Poisson Process using $\lambda_{P}$ as the mean rate.

(iii) Choose randomly a connection weight between 0 and 1 for each spike in the spike train.

(iv) Assign randomly a sign for each connection weight with equal positive and negative possibility.

(v) Calculate the input currents for the generated spike train by using (3.13).

(vi) Solve numerically the differential equation in (3.6) with ionic current defined in (3.14) for 1000 ms as in the Wolfram Mathematica computation environment.

(vii) Record membrane voltage $u$, gating variables $m$, $n$, and $h$ at 0.1 ms time steps and save them to a file.

The experiment was repeated 1000 times with different mean rates $\lambda_{P}$ to ensure statistically reliable results.
3.2.2 Firing Thresholds

There are published methods to estimate the firing threshold of a biological neuron [80], yet they are all based on constant stimulations. In our experiment, the threshold could be estimated statistically when the neuron is driven by random spike trains, which is more realistic, especially when a neuron is the fundamental element of a large neural network.

The idea for estimating the threshold statistically is based on the assumption that all action potential spikes are similar and have only one peak in each spike. Since all non-spiking peaks have the peak membrane voltage much lower than those of the action potential peaks, no local maxima should be found between the threshold $u_{th}$ and the average action potential peak value. We searched for all local maxima in the recorded membrane voltage and calculated the distribution of probability density of the local maximum values, which was plotted in Figure 3.2(a). The probability density curve shows that most local maxima fell into either the range from $-20 \text{ mV}$ to $10 \text{ mV}$, or in the range from $90 \text{ mV}$ to $110 \text{ mV}$. Since there are almost no peaks in the voltage region between $10 \text{ mV}$ and $90 \text{ mV}$, the threshold voltage could be assumed to be below $50 \text{ mV}$.

We defined the firing threshold as a membrane voltage below which 95% of all recorded regular peaks could be found:

$$u_{th} : P_{peaks} \left\{ U | U < u_{th} \right\} = 0.95.$$  \hspace{1cm} (3.15)
Figure 3.2 The firing threshold of the membrane voltage. (a) Probability distribution of the local maximums of membrane voltage. The left region of this distribution indicates those peaks did not induced action potentials; the right region of this distribution shows the action potential peaks. Almost no local maximum could be found in between; (b) Cumulative distribution of the regular peaks; the firing threshold $u_{th}$ is defined so that 95% of regular peaks happens below it.
Since there were no peaks located between 10 mV and 90 mV, all peaks lower than 50 mV were considered as regular peaks. The cumulative distribution function (CDF) of the probability density for these regular peaks were calculated and plotted in Figure 3.2(b). The firing threshold $u_{th} = 4.69$ mV was found from the CDF of the regular peaks.

### 3.2.3 Kernel Functions

The variable conductance for the two ion channels were calculated at discrete time steps $t[i]$ for $i = 1, 2, \ldots, k$ as follows:

\[
g_{Na}[i] = g_{Na} m[i]^3 h[i] \\
g_k[i] = g_k n[i]^4.
\]  

In order to find the kernel functions, the simulation time $t[i]$ should be converted into survival time. We searched for all the firing moments with step index $f_1, f_2, \ldots, f_p$ satisfying:

\[
\Theta(u[i] - u_{th}) - \Theta(u[i-1] - u_{th}) = 1,
\]  

which are all the moments when conditions $u[i-1] < u_{th}$ and $u[i] \geq u_{th}$ are satisfied. The survival time for each step was calculated as

\[
\tau[i] = t[i] - t[f_j], \\
\text{when } t[f_j] \leq t[i] < t[f_{j+1}],
\]  

for all $j = 2, 3, 4, \ldots, p$. A boundary condition $f_{p+1} > 1000$ ms was used to calculate the survival time for steps after the last action potential in each experiment, yet the recording
starts from \( t[f] \) because those time steps before the first action potential in each experiment should be discarded as they have no corresponding survival times.

The membrane voltage and conductance were categorized into sequential survival time intervals:

\[
[\tau_0, \tau_1), [\tau_1, \tau_2), ..., [\tau_g - 1, \tau_g).
\]

The mean values and standard variances were calculated for \( u, g_{Na}, \) and \( g_K \) at each

![Graph](image)

Figure 3.3 Mean value of Membrane voltage (black squares using left and bottom axes) and conductance of sodium and potassium channel (red rounds and green triangles using right and top axes) versus survival time. The standard variances are indicated by the error bars at all sample points individually.
interval, with results shown in Figure 3.3.

Although they vary slightly from their mean values, the sodium and potassium channel conductance and the membrane voltage evolve along certain trajectories after each firing of the action potential. We could find that the variances of membrane voltage are negligible after the spike, due to the disturbance originated from random stimulations. However, the small variances of ionic conductance indicate that the ionic conductance repeats precisely for each action potential, and is insensitive to the input stimulations. The mean values of $g_{Na}$ and $g_{K}$ could then be used to construct the $g_{kern}$ and $i_{kern}$ functions, respectively.

The integration of $g_{kern}$ and $i_{kern}$ are more important than the conductance in affecting the dynamics of membrane voltage, thus we would like to fit the numerically integrated $g_{Na}$ and $g_{K}$ data to the integral functions of $f_{i}$. Suppose the integrated conductance of sodium and potassium over the survival time can be formulated as:

$$
\Gamma_{Na}(\tau) = \int_{0}^{\tau} g_{Na}(t)dt
$$

$$
\Gamma_{K}(\tau) = \int_{0}^{\tau} g_{K}(t)dt.
$$

(3.20)

The following sigmoid functions are used to fit $\Gamma_{Na}$ and $\Gamma_{K}$ data for the construction of the kernel functions:

$$
S_{x}(\tau) = \frac{A_{x}}{1 + e^{(-\tau - \lambda_{x})/\tau_{x}}} + k_{x} \tau,
$$

(3.21)
where subscript $x$ could be “Na” or “K” indicating that the function is fitted to the integrated sodium or potassium conductance data. The fitted parameters for these functions are shown in Table 3.1. The fitting of function $\Gamma_{Na}(\tau)$ to the integrated $g_{Na}$ data yields SSE = 34.91 with adjusted R-square = 0.9977 and RMSE = 0.7386; while the fitting of function $\Gamma_{K}(\tau)$ to the integrated $g_{K}$ data yields SSE = 63.65 with adjusted R-square = 0.9983 and RMSE = 0.9972, both of which show that the fittings are good and are made with high confidence.

Table 3.1 Fitting parameters for the kernel functions

<table>
<thead>
<tr>
<th>$x$</th>
<th>$A_x$ (mS·ms/cm²)</th>
<th>$l_x$ (ms)</th>
<th>$u_x$ (ms)</th>
<th>$k_x$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>35.88</td>
<td>0.3180</td>
<td>2.128</td>
<td>0.0115</td>
</tr>
<tr>
<td>K</td>
<td>39.53</td>
<td>0.7889</td>
<td>3.837</td>
<td>0.3118</td>
</tr>
</tbody>
</table>

The variable conductance and bias current function needed in (3.1) could be defined in the following way:

\[
g_{\text{kern}}(\tau) = k_L + \frac{dS_{Na}}{d\tau} + \frac{dS_{K}}{d\tau}
\]

\[
i_{\text{kern}}(\tau) = i_L + \frac{dS_{Na}}{d\tau}E_{Na} + \frac{dS_{K}}{d\tau}E_{K},
\]

where $k_L$ is the constant leaky conductance and $i_L$ is the bias current contributed by the constant leaking channel defined as follows:

\[
i_L = k_L E_L.
\]
The constants $E_{Na} = 115$ mV, $E_K = -12$ mV and $E_L = 10.6$ mV are reversal potentials, the same as those found in the HH model. We could select the linear leaky conductance $g_L = 0.3 \text{ mS/cm}^2$ suggested by HH model as the constant $k_L$ in (3.23), yet there would be a small amount of mismatch which yields a non-zero $i_L = 1.1013 \mu\text{A/cm}^2$ in that case. Thus we adjusted $k_L$ to 0.1961 mS/cm$^2$ so that the bias current vanishes when the neuron stays quiescent.

### 3.2.4 GLIF Model Solutions

According to Appendix, the ODE of GLIF model could be solved analytically to yield the following:

$$u(t_r) = u_{\text{kern}}(t_r) + u_{\text{syn}}(t_r),$$

where $t_r$ is the survival time with respect to the most recent firing time $t_f$, $u_{\text{kern}}$ is the action potential membrane voltage trajectory and $u_{\text{syn}}$ is the contribution of input current spikes to the membrane voltage.

Entities $u_{\text{kern}}$ and $u_{\text{syn}}$ are derived using the following equations:

$$u_{\text{kern}}(t_r) = \Pi(t_r)\left[u_{\text{th}} + \int_0^{t_r} \frac{i_{\text{kern}}(\tau)}{C_m \Pi(\tau)} d\tau\right]$$  \hspace{1cm} (3.25)

$$u_{\text{syn}}(t_r) = \Pi(t_r)\int_0^{t_r} \frac{i_{\text{syn}}(\tau)}{C_m \Pi(\tau)} d\tau,$$  \hspace{1cm} (3.26)

with the function $\Pi(t_r)$ defined as:

$$\Pi(t_r) = e^{-\int_0^{t_r} \frac{g_{\text{kern}}(\xi)}{C_m} d\xi}.$$  \hspace{1cm} (3.27)
The GLIF ODE solution is still difficult to implement given its inherent computational complexity. The function $\Pi$ carries with it a heavy computational burden and is practically impossible to be simplified by means of table lookups. Although the function value decreases to zero within several milliseconds, a divide by zero error may occur in (3.25) and (3.26) if the tail of function $\Pi$ at large $t_r$ is trimmed to zero.

Our method discretized the ODE solution and made some further simplifications so that the calculation of the $\Pi$ function could be substituted by table lookup operations.

The first simplification was to substitute the linear raising and exponentially decaying current shape function $i_s^*$ by a pulse shape current function as follows:

\[
i_{s,\text{pulse}}(\tau) = \frac{\Delta u_s}{\Delta s} P_{\text{unit}}(\tau, \Delta s)
\]

\[
P_{\text{unit}}(t_0, \Delta t) = \left[ \Theta(t_0) - \Theta(t_0 + \Delta t) \right], \tag{3.28}
\]

where $P_{\text{unit}}$ is a unit pulse function, while $\Delta s$ and $\Delta u_s$ are the width and height of the pulse, respectively. The membrane voltage is increased by $\Delta u_s$ when such pulse stimulations arrive. We set $\Delta u_s = 8.96 \text{ mV}$ in the proposed method, which is similar to the maximum possible membrane voltage that a GLIF neuron could generalize after it is stimulated by a single input spike sent from a synapse fully connected to the neuron, when the neuron is in quiescent status.

Using (3.28), we could rewrite (3.26) as:
\[ u_{\text{syn}}(t) = \int_0^t \frac{\Pi(t)}{\Pi(\tau) \Delta s} I_s d\tau, \quad (3.29) \]

where \( I_s \) is the overall input current defined as:

\[ I_s = \sum_j w_j P_{\text{unit}}(\tau - s^{(j)}, \Delta s) \quad (3.30) \]

Since no singular points could be found from the integrand function in (3.29), we could use summations at small time intervals \( \Delta s \) for \( \tau \) to approximate the integration:

\[ u_{\text{syn}}[n] = \sum_{p=0}^{n} \frac{\Pi(n\Delta s)}{\Pi(p\Delta s)} \Delta u_s \sum_j w_j \delta_{p,f_j}, \quad (3.31) \]

where \( n \) is the current time index, \( \delta \) denotes the Kronecker delta function, \( p \) is the summation index and \( f_j \) is the \( j \)-th input spike time index defined by:

\[ n\Delta s \leq t < (n+1)\Delta s \]

\[ f_j\Delta s \leq s^{(j)} < (f_j + 1)\Delta s. \quad (3.32) \]

Therefore, the discrete solution of the GLIF ODE can be written as follows:

\[ u[n] = u_{\text{kern}}[n] + u_{\text{syn}}[n]. \quad (3.33) \]

where \( u_{\text{kern}}[n] \) is the discrete version of \( u_{\text{kern}}(t) \):

\[ u_{\text{kern}}[n] = u_{\text{kern}}(n\Delta s) \quad (3.34) \]

### 3.2.5 Exponential Moving Average Implementation

The Exponential Moving Average (EMA) is a way to calculate the moving average \( S \) for a set of discrete data \( x \):
\[ S[0] = x[0] \]
\[ S[n+1] = \alpha S[n] + \beta x[n+1]. \] (3.35)

The average value \( S \) could then be expressed as:
\[ S[n] = \alpha^n x[0] + \beta \sum_{i=1}^{n} e^{(n-i)\log\alpha} x[i], \] (3.36)
so that when \( \alpha < 1 \), the influence of inputs decrease exponentially from the latest sample \( x[n] \) to earlier samples. For normal EMA, parameters \( \alpha \) and \( \beta \) are constants and fulfill boundary condition \( \alpha + \beta = 1 \), yet the exponential averaging properties are still satisfied as long as \( \alpha < 1 \). We henceforth denote EMA with variable averaging parameters as the Generalized EMA (GEMA).

In our model solution, we could thus define
\[ \alpha[n] = \frac{\Pi(n \Delta s)}{\Pi[(n-1) \Delta s]} = e^{\int_{(n-1)\Delta s}^{n\Delta s} \frac{g_{\text{syn}}(\xi)}{C_{\text{ne}} d\xi}} \] (3.37)
\[ \beta = 1, \]
so that the discrete solution of GLIF ODE could be written in an iterative fashion:
\[ u[n] = u_{\text{ker}}[n] + u_{\text{syn}}[n] \]
\[ u_{\text{syn}}[n] = \alpha[n] u_{\text{syn}}[n-1] + \Delta u \sum_{j} w_j \delta_{n,j} \] (3.38)
\[ u_{\text{syn}}[0] = 0. \]

Note that since \( g_{\text{ker}} \) is smooth and \( \Delta s \) is small, we have
\[ \alpha[n] = e^{-g_{\text{syn}}(\pi \Delta s)/C_{\text{ne}}}, \] (3.39)
defined as the variable averaging parameter.
The variable parameter $\alpha$ has been plotted in Figure 3.4 for a simulation step size $\Delta s = 0.1$ ms, where we could find that $\alpha$ is always smaller than one, satisfying the EMA requirement. It can also be observed that $\alpha$ tends to be constant at larger values of $n$, indicating that the input spikes are regularly exponential moving averaged into $u_{syn}$ during that period. Yet in the “Variable Window Region”, $\alpha$ is significantly decreased. Since $\alpha$ is the parameter indicating the averaging window size of the GEMA, a smaller $\alpha$ means that less input spikes are averaged into $u_{syn}$. The input spikes arriving at the neuron in the variable window region will be more difficult to increase the membrane voltage and less likely to fire an output spike than would the input spikes which arrive in the constant window region. This shows that GLIF model could characterize the relative refractory period as it appears in biological neurons and the HH model. In contrary, the

Figure 3.4 GEMA window width according to the survival time. The simulation step size is 0.1 ms
averaging parameter $\alpha$ is always a constant if the same GEMA process is repeated for the NLIF model, proving that no relative refractory period could be found in the NLIF model.

### 3.3 GLIF Model Performance

The two major issues that define a model’s performance are its speed and resulting accuracy. The less computation effort to simulate a single neuron, the easier it is to integrate it into a large-scale ASNN.

#### 3.3.1 GLIF Model Accuracy

Accuracy is a measure that defines how reliable a model could be in reproducing the behavior of a biological neuron. The computation effort of a neuron model could be easily evaluated by simulating it together with another known model under the same conditions, and then compare their computation time. Yet the evaluation of the accuracy for a model is a bit more complicated. Since no quantitative measurement exists to evaluate the accuracy of a neuron model so far. In this study, the Missed Fire Rate (MFR) and the Accidental Fire Rate (AFR) are used as indicators of a neuron model’s accuracy.

Suppose one input spike train is used to stimulate both a reference neuron model (a model considered as reliable in reproducing the biological neuron activities) and a testing model. Their respective generated output spike trains are $S_r = \{s^{(1)}_r, s^{(2)}_r, \ldots, s^{(m)}_r\}$ and $S_t = \{s^{(1)}_t, s^{(2)}_t, \ldots, s^{(m)}_t\}$, which means the reference model fires output spikes at simulation time $s^{(i)}_r$ and the testing model fires at simulation time $s^{(i)}_t$, where
\( i = 1, 2, \ldots, m \) and \( j = 1, 2, \ldots, n \) are the spike indices. A small tolerance \( \varepsilon \) is used so that if a spike pair \( s_r^{(i)} \) and \( s_t^{(j)} \) was found satisfying \( |s_r^{(i)} - s_t^{(j)}| < \varepsilon \), the \( i \)-th spike in \( S_r \) and the \( j \)-th spike in \( S_t \) should be considered as matched spikes.

By searching all the matched spike pairs throughout the two output spike trains, and removing the matched spikes from their own output spike trains immediately after they are found, the remaining spikes in \( S_r \) are called the Missed Fires, and the remaining spikes in \( S_t \) are called the Accidental Fires. Note that the removing operation of matched fires from their own spike trains ensured that each spike in either output spike trains could be matched to no more than one spike in the other output spike train.

We could define the error rate of missed fires and accidental fires as

\[
\text{MFR} = \frac{N_{MF}}{N_{\text{match}}}, \quad \text{AFR} = \frac{N_{AF}}{N_{\text{match}}},
\]

(3.40)

where \( N_{MF} \) is the number of missed fires, \( N_{AF} \) is the number of accidental fires and \( N_{\text{match}} \) is the number matched output spike pairs.

MFR and AFR could indicate how well the testing model performs in making errors of missed fires or accidentally fires. However, we do need a single measurement to estimate the overall accuracy of the testing model. Following the definitions used in classification studies, we could easily define the matched fires, missed fires and accidental fires as “True Positives”, “False Positives” and “False Negatives” respectively, yet the “True
Negatives” are hard to define for this case and thus it is hard to use the sensitivity and specificity measurements to estimate the model accuracy. However, the F\textsubscript{1} score defined as:

\[
F_{1} = \frac{2N_{\text{match}}}{2N_{\text{match}} + N_{\text{MF}} + N_{\text{AF}}},
\]

(3.41)
could be used instead as a good measurement of the overall accuracy of the testing model, which will decrease to zero when no match pair is found and increase to one when all spikes are paired between the output spike trains from the testing model and the reference model.

We have built 100 neurons using HH model, NLIF model and GLIF model respectively in Matlab environment on an Intel i7-2600 workstation with 4GB memory. The HH model described in (3.14) was considered as the reference model due to its widely accepted accuracy in reproducing the activities of the giant axon neuron in a squid. The GLIF parameters and kernel functions were extracted from the HH model, using the methods described in section 3.2. NLIF neurons were also built to provide a comparison with GLIF neurons in terms of the accuracy and speed. The leaky resistance \( R = 5.2 \, \text{k}\Omega \cdot \text{cm}^2 \) and time constant \( \tau_m = 5 \, \text{ms} \) were calculated by using (3.12) for NLIF neurons. An absolute refractory period \( \Delta_{\text{abs}} \) as suggested by Schliebs \textit{et al.} [82] was adopted for both neuron models to eliminate membrane voltage oscillations.

In this experiment, 100 spike trains with uniform length of 1000 ms were generated as neuron inputs by Poisson Process with mean frequency \( \lambda_P = 150 \, \text{Hz} \). The HH neurons
were simulated for 1000 ms and their output firing times were recorded as reference outputs. The NLIF neurons and GLIF neurons were also simulated for 1000 ms under same stimulation. The F₁ scores were calculated for both NLIF and GLIF neurons by comparing their output spike trains with the output spike train from HH neurons.

Since the length of absolute refractory period $\Delta^{\text{abs}}$ and the threshold level $u_{\text{th}}$ in NLIF model is very important to the model accuracy, we have repeated the above experiment at different combinations of $\Delta^{\text{abs}}$ and $u_{\text{th}}$: the values of $\Delta^{\text{abs}}$ and $u_{\text{th}}$ were linear spanned in the range of 0 ms to 20 ms and from 2 mV to 20 mV. After the F₁ scores for all pairs of $\Delta^{\text{abs}}$ and $u_{\text{th}}$ have been obtained, a 2-D interpolation was performed on those F₁ scores so that the combination which provides the best F₁ score could be found numerically.

Although $\Delta^{\text{abs}}$ is not required for GLIF model and $u_{\text{th}}$ has been estimated theoretically in section 3.2.2, we still performed such accuracy tests on variable $\Delta^{\text{abs}}$ and $u_{\text{th}}$ values for GLIF model for the purpose of estimating the influence of these two parameters on the GLIF model accuracy.
3.3.2 GLIF Model Efficiency

Figure 3.5 F1 scores at different firing thresholds and absolute refractory period lengths. (a) F1 scores for NLIF model; (b) F1 scores for GLIF model.
Besides the accuracy, the speed of HH model, NLIF model and GLIF model have also been evaluated by building 1000 neurons of each model in Matlab environment and recording the simulation time for each neuron model. In order to make sure that the numerical solution of the ODEs model could converge in the simulation, forth-order Runge-Kutta (RK4) method was adopted for HH model and NLIF model, while GEMA method was used for GLIF model. The experiment was performed at variable simulation step size with calculation times recorded, and with MFR and AFR evaluated for each step size.

3.4 Results and Discussion

The accuracy of NLIF model and GLIF model depends on the threshold \( u_{th} \) and absolute refractory period \( \Delta_{abs} \) as shown in Figure 3.5. The outputs from HH neurons were used as the reference output spike train, and a matching tolerance \( \epsilon = 5 \) ms was adopted to find the matched spike pairs between the testing model and the HH model.

The \( F_1 \) scores of NLIF model depend on both \( u_{th} \) and \( \Delta_{abs} \) values. Lower \( u_{th} \) and \( \Delta_{abs} \) settings could increase the chances of accidental fires while higher \( u_{th} \) and \( \Delta_{abs} \) could increase the chances of missed fires, both of which could bring down the \( F_1 \) score for the NLIF model. The best \( F_1 \) score for the NLIF model was obtained at \( u_{th} = 8.99 \) mV and \( \Delta_{abs} = 15.25 \) ms. We could also find from Figure 3.5(b) that the \( F_1 \) score of GLIF model exclusively depends on the threshold, and is not sensitive to \( \Delta_{abs} \), especially when \( \Delta_{abs} \) is shorter than 16 ms. The best \( F_1 \) score was found at \( u_{th} = 3.66 \) mV and \( \Delta_{abs} = 5.35 \) ms.
A portion of the recorded membrane voltage and firing times of GLIF neurons, HH neurons and NLIF neurons under same stimulus are shown in Figure 3.6.

Figure 3.6 Comparison of membrane voltage evolution under current input using a Poisson input spike train for NLIF, GLIF and HH models. Vertical bars marked under the membrane voltage of NLIF model indicate its spiking locations.
Compared to the NLIF model, the GLIF model could reproduce similar spike shape action potentials as the HH model. It could be found intuitively from Figure 3.6 that the GLIF model generates fewer accidental fires than the NLIF model. The only fire missed by both GLIF and NLIF models in Figure 3.6 is the second fire of two adjacent fires found in the HH model, which is probably due to the bursting fire mechanism under large stimulation.

Such bursting feature is impossible to be captured by any one-dimensional neuron models because of the constant absolute refractory period applied to these models. The constant absolute refractory period prohibits the neuron to fire multiple action potentials in short intervals, while in biological neurons and high-dimensional neuron models, large stimulation could significantly change the refractory behavior of the neuron and provide a chance for the neuron to fire bursting spikes.

A detailed comparison of the membrane potentials for these three models is shown in Figure 3.7, in which we could find that the membrane potentials are almost the same for these three models when the survival time is longer than 10 ms (in the so-called “silent period”), proving that the NLIF model is a good approximation when the neuron stays quiescent for long enough. Yet the membrane voltage trajectory of NLIF model is completely different from GLIF and HH models in the post-fire region, marked as the refractory period in Figure 3.7.
In GLIF and HH model, the influence of the input current to membrane voltage is attenuated at the post fire region, i.e., the relative refractory period, which makes the neuron less capable yet not impossible to fire new action potentials during that period. However, in NLIF model, the leaky resistance is a constant and the influence of input spikes is always the same no matter if the neuron fires or not. Although the absolute refractory period assumed for the NLIF model could serve as a coarse approximation of the gradual changed input sensitivity found in the biological neuron, and the best accuracy could be found when $\Delta^{\text{abs}}$ has been adjusted similar to the length of relative refractory period found in HH model, the overall accuracy of NLIF is still low because that the “all-or-nothing” simplification of the absolute refractory period is not feasible in

![Figure 3.7 Zoom view of simulated membrane potentials in GLIF, HH and NLIF models.](image)
reproducing the input sensitivity variations after the neuron fires. In contrary, the time-
dependent averaging parameter $\alpha$ extracted from HH model provides a much better
approximation of the input sensitivity during refractory period, thus the accuracy of GLIF
model, even without assumption of absolute refractory period, is much better than NLIF
model at the post-fire period.

The speed of GLIF model was also tested and compared to HH and NLIF models with
results shown in Table 3.2. Solving for the HH model at step sizes larger than 0.08 ms
failed, because the RK4 method failed to converge at some steps while solving the ODE
model under those step sizes. Generally, the GLIF model with GEMA method is about 30%
faster than the NLIF model, while both of these LIF models are about 50 times faster than
the HH model.

The accuracy measures of MFR and AFR have also been evaluated at various step sizes
for both NLIF and GLIF neurons, with results plotted in Figure 3.8. It can be observed
that both AFR and MFR of GLIF are much lower than those of the NLIF model, and are
empirically found to be independent of the simulation step size.

Comparatively, the MFR of the NLIF model increases dramatically when the simulation
step size is larger than 0.4 ms, which is an indication that the errors of RK4 method
increase along with the step size. Since the computation time of either models decrease
exponentially while the step size increase, the possibility of selecting a larger step size for
the GLIF model makes it more competitive in large-scale ASNN applications.
Table 3.2 Computation time of HH, NLIF and GLIF models under various simulation steps.

<table>
<thead>
<tr>
<th>$h$ (ms)</th>
<th>$T_{HH}$ (s)</th>
<th>$T_{NLIF}$ (s)</th>
<th>$T_{GLIF}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>100.1</td>
<td>3.554</td>
<td>2.574</td>
</tr>
<tr>
<td>0.04</td>
<td>50.04</td>
<td>1.775</td>
<td>1.288</td>
</tr>
<tr>
<td>0.06</td>
<td>33.36</td>
<td>1.180</td>
<td>0.8588</td>
</tr>
<tr>
<td>0.08</td>
<td>25.02</td>
<td>0.8881</td>
<td>0.6453</td>
</tr>
<tr>
<td>0.10</td>
<td>N/A</td>
<td>0.7125</td>
<td>0.5143</td>
</tr>
<tr>
<td>0.20</td>
<td>N/A</td>
<td>0.3596</td>
<td>0.2577</td>
</tr>
<tr>
<td>0.30</td>
<td>N/A</td>
<td>0.2431</td>
<td>0.1711</td>
</tr>
<tr>
<td>0.40</td>
<td>N/A</td>
<td>0.1858</td>
<td>0.132</td>
</tr>
<tr>
<td>0.50</td>
<td>N/A</td>
<td>0.1481</td>
<td>0.1032</td>
</tr>
<tr>
<td>0.60</td>
<td>N/A</td>
<td>0.1246</td>
<td>0.0863</td>
</tr>
<tr>
<td>0.70</td>
<td>N/A</td>
<td>0.1094</td>
<td>0.0734</td>
</tr>
<tr>
<td>0.80</td>
<td>N/A</td>
<td>0.0954</td>
<td>0.0646</td>
</tr>
<tr>
<td>0.90</td>
<td>N/A</td>
<td>0.0844</td>
<td>0.0572</td>
</tr>
<tr>
<td>1.00</td>
<td>N/A</td>
<td>0.0766</td>
<td>0.0517</td>
</tr>
</tbody>
</table>

Figure 3.8 Missed Fire Rates and Accidental Fire Rates of NLIF and GLIF model under various simulation step size.
4. WAVELET ENCODING METHOD FOR ASNN

In this section, we will demonstrate that an array of specially designed LIF neurons could perform wavelet decomposition of temporal signals. This special design of a LIF neuron differs from traditional LIF neurons by incorporating a two-stage spike triggered modulate-and-integrate module to pre-process the input signal. Such design was inspired by the multiplication relationship found among afferent synaptic currents in biological neurons [83]. Delay synchronized spikes sent to the two synapses integrated in the special designed LIF neuron could trigger the wavelet transform of the input signal at certain time scales, and encode the spectrum amplitudes into delays between the output fire times and the control spike arriving times. Simulations in this research were conducted using NEural Simulation Tool [84] (NEST) with custom made neuron models.

4.1 Encoding with LIF Neurons

Spiking neuron models are typically a set of Ordinary Differential Equations (ODE) which attempt to capture the dynamics of the neuron membrane potential. Different neuron models have been proposed by researchers to mimic the electrical behaviors of biological neurons. Among these neuron models, LIF model was believed to be a reasonable simplification of biological neuron with balanced accuracy and efficiency. LIF spiking neuron is described by one-dimensional ODE using the following equations:

\[ \tau \frac{du(t)}{dt} = -u(t) + \frac{\tau}{C_m} I_{all}(t) \]
where \( u \) is the membrane potential, \( \tau \) and \( C_m \) are the time constant and capacitance of the neuron, respectively, with \( I_{\text{all}} \) defining the overall afferent current. The firing condition and post-fire behavior of the LIF neuron in (4.1) can be defined by the following equation:

\[
\text{if } u = u_{\text{th}} \text{ and } \frac{du(t)}{dt} > 0, \quad u \leftarrow u_{\text{c}}
\]

(4.2)

where \( u_{\text{th}} \) is the firing threshold and \( u_{\text{c}} \) is the post-fire resetting potential. Note that a derivative condition is applied to the firing conditions in the same manner as in Wang et al. [85]. Such derivative condition ensures that the neuron only fires when its membrane potential in an upward trend crosses the threshold, a condition which is thus set to avoid accidental fires if the resting potential of the neuron is higher than its firing threshold.

The stimulation to LIF neuron is typically assumed to be a summation of all weighted synaptic currents and an external current:

\[
I_{\text{all}}(t) = I_e(t) + \sum_j w_j I_s(t - s_j)
\]

(4.3)

In this equation, \( I_e(t) \) is the external current, \( I_s(t) \) is the shape function of the post-synaptic current (PSC), \( s_j \) is the time that the \( j \)-th spike arrives at the synapse, and \( w_j \) is the connection efficacy corresponding to the \( j \)-th input spike.

Consider a quasi-static input signal being used as the external current to the LIF neuron, and no synaptic stimulation was connected, (4.1) could be solved as:
\[ u(t) = u_e \exp \left( -\frac{t - t^f}{\tau} \right) + \tau I_e(t) \left[ 1 - \exp \left( -\frac{t - t^f}{\tau} \right) \right] / C_m \]  \hspace{1cm} (4.4)

where \( t^f \) is the most recent fire time of the LIF neuron. The output spike interval \( T \) is thus a function of \( I_e \) as defined below

\[ T = \tau \ln \left( \frac{u_e C_m - \tau I_e}{u_{\text{th}} C_m - \tau I_e} \right) \]  \hspace{1cm} (4.5)

Since the reset potential is usually lower than the threshold \( u_{\text{th}} \), larger \( I_e \) yields shorter spike interval and thus higher firing rate over a short time window. The input signal is rate encoded in this configuration.

Rumbell et al. [66] suggested a method to generate phase encoded spike train using LIF neurons. A global inhibitory neuron has been connected to all encoding neurons, so that the reset times of these neurons are synchronized, and the firing time interval found in (4.5) could be converted into the firing delays between neurons.

Encoding methods using LIF neurons however suffer from one major drawback in that the input signal should be quasi-static in comparison to the time constant of the LIF neuron. Although temporal decomposition methods such as Gaussian receptors could reduce the fluctuation of the input signal, the number of receptors increase dramatically with increasing frequency of the input signal, which prevents the encoding method from capturing fast transients in the input signals.
4.2 Spike Triggered Modulation

Although linear summation of synaptic currents and external current as performed in (4.3) has been widely accepted as a simplified relationship among the afferent stimulations in large-scale ASNN, the interaction between post-synaptic currents was found to be more complicated in biological nervous system. Koch and Segev [83] found that biological neurons might approximate sum of products among different groups of synaptic currents. Inspired by this finding, we designed a two-stage modulate-and-integrate module, where the multiplication is performed instead of summation between the input signal and synaptic currents. The first stage of the module incorporates the integration of the multiplication of external current and a wavelet shape synaptic current, while the second stage modulate the output from first stage with an exponential decay synaptic current. We will prove that using our preprocessing module together with a LIF neuron, input signal could be decomposed into wavelet spectrum and such spectrum amplitude could be encoded into synchronized spike trains.

In reference to Figure 4.1, \( C_{\text{int}} \) and \( C_{\text{enc}} \) are delay synchronized clock spikes satisfying:

\[
i_{i}^{\text{enc}} - i_{i}^{\text{int}} = T_{e}
\]  

(4.6)

where \( T_{e} \) is the delay phase, \( i_{i}^{\text{int}} \) and \( i_{i}^{\text{enc}} \) are time of spikes in \( C_{\text{int}} \) and \( C_{\text{enc}} \) respectively, with \( i = 1,2,\ldots,n \) being the index of each spike. The interval of spikes in both \( C_{\text{int}} \) and \( C_{\text{enc}} \) is \( T_{\text{clk}} \). \( C_{\text{int}} \) and \( C_{\text{enc}} \) are converted into post-synaptic current \( I_{\text{enc}} \) and \( I_{\text{int}} \) by synapse \( S_{\text{int}} \) and \( S_{\text{enc}} \) respectively. Input signal \( I_{e} \) is multiplied with \( I_{\text{int}} \), and integrated by neuron
\( N_{\text{int}} \) into its state variable \( v \). \( N_{\text{enc}} \) is a normal LIF neuron, stimulated by the absolute amplitude of \( v \) modulated with \( I_{\text{enc}} \).

The overall dynamics of this encoding unit could be specified by the following equations:

\[
\tau \frac{du(t)}{dt} = -u(t) + \frac{\tau}{C_m} |v(t)| I_{\text{enc}}(t) \tag{4.7}
\]

\[
a \frac{dv(t)}{dt} = I_z(t) I_{\text{int}}(t) \tag{4.8}
\]

where \( u \) is the state variable of \( N_{\text{enc}} \), \( I_{\text{enc}} \) and \( I_{\text{int}} \) are summations of the post-synaptic currents of spikes in \( C_{\text{enc}} \), and \( C_{\text{int}} \) respectively, and are defined as follows:

\[
I_{\text{enc}}(t) = \sum_i \exp\left(-\frac{t-t_{i_{\text{enc}}}}{\tau}\right) \Theta(t-t_{i_{\text{enc}}}) \tag{4.9}
\]

\[
I_{\text{int}}(t) = \sum_i \sqrt{a} \Psi(t-t_{i_{\text{int}}}-d, \sigma) \Theta(t-t_{i_{\text{int}}}) \tag{4.10}
\]

where \( \Psi \) is a wavelet mother function used as the PSC for \( S_{\text{int}} \), with \( a \) representing the scale of the wavelet, \( \sigma = af_s \) indicating the time scale of the wavelet related to the Figure 4.1 Structure of the Two-Stage Modulate-and-Integrate Module
sampling frequency $f_s$, $d$ serving as an offset parameter, and $\Theta$ being a Heaviside step function. We selected a shifted Mexican-hat wavelet mother function for $\Psi$ as a demonstration here:

$$
\Psi(t, \sigma) = \frac{2}{\sqrt{3\pi^{1/4}}} \left(1 - \frac{t^2}{\sigma^2}\right) \exp\left[-\frac{t^2}{2\sigma^2}\right]
$$

Assuming that the length of integration period $T_i$ satisfies $T_i < T_{\text{clk}}$, we could define $d = T_i / 2$ in (4.10), so that the wavelet function is centered within each integration window. The concept of wavelet mother function will be discussed in section 4.3.2.

**Reference source not found.** Note that both $I_{\text{enc}}$ and $I_{\text{int}}$ are constructed in a unitless manner for the model simplification.

Suppose that each spike in $C_{\text{int}}$ could reset the state variable $v$ of neuron $N_{\text{int}}$ to zero, and that $\sigma$ is significantly smaller than $T_{\text{clk}}$, then (4.7) could be solved for $t_{i}\leq t < t_{i+1}$ as:

$$
v(t) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} I_c(\zeta) \Psi(\zeta - t_{i\text{int}} - T_i / 2, \sigma) d\zeta
$$

Suppose further that $\sigma$ is significantly smaller than $T_i$, and consider that $\Psi(t, \sigma) \to 0$ when $t > T_i$ or if $t < 0$, then (4.12) could be approximated by:

$$
v(t) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} I_c(\zeta) \Psi(\zeta - t_{i\text{int}} - T_i / 2, \sigma) d\zeta
= X_w \left(t_{i\text{int}} + T_i / 2, \sigma\right)
$$
for \( t_{i_{\text{int}}}^{\text{int}} + T_i \leq t < t_{i+1}^{\text{int}} \), where \( X_w \) is the wavelet transform of input \( I_e \) at translation \( t_{i_{\text{int}}}^{\text{int}} + T_i / 2 \) and time scale \( \sigma \).

Assuming that:

\[
T_i < T_e < T_{\text{elk}}
\]  \hspace{1cm} (4.14)

and suppose each input spike from \( C_{\text{enc}} \) could reset the state variable from \( u \) to \( u_c \) for neuron \( N_{\text{enc}} \), (4.7) could be solved for \( t_{j_{\text{enc}}}^{\text{enc}} \leq t < t_{j+1}^{\text{enc}} \) as:

\[
u(\Delta t) = u_c \exp\left(-\frac{\Delta t}{\tau}\right) + V(\Delta t)
\]  \hspace{1cm} (4.15)

\[
V(\Delta t) = \frac{\tau \Delta t}{C_m} \exp\left(-\frac{\Delta t}{\tau}\right) |X_w\left(t_{j_{\text{int}}}^{\text{int}} + T_i / 2, \sigma\right)|
\]  \hspace{1cm} (4.16)

where \( \Delta t \) is the elapsed time since last input spike from \( C_{\text{enc}} \) arrives at the neuron. Note that the absolute value operation applied to \( v \) makes \( V(\Delta t) \) a function of the absolute spectrum of the wavelet transform \( X_w \). The absolute spectrum is preferable to power spectrum of the wavelet transform, in the sense that it ensures that the units in equation (4.15) are balanced without need for extra constants.

We considered two different combinations of reset potential \( u_c \) and output firing threshold \( u_{\text{th}} \) for \( N_{\text{enc}} \):

(i) Negative threshold: \( u_c < u_{\text{th}} < 0 \).

(ii) Positive threshold: \( u_c = 0 \) and \( u_{\text{th}} > 0 \).

In the first combination, as long as
\[ T_{\text{clk}} - T_c > \tau \ln \left( \frac{u_{\text{th}}}{u_c} \right) \]  

(4.17)

\( V(\Delta t) \) is a non-negative function. The membrane potential will exceed the threshold and an output spike will be generated during each time segment \([t_{i}^{\text{enc}}, t_{i+1}^{\text{enc}}]\). The fire delay \( T \) in the \( i \)-th segment could be solved from:

\[
|X_w| = \frac{C_m}{\tau T} \left[ u_{\text{th}} \exp \left( \frac{T}{\tau} \right) - u_c \right]
\]  

(4.18)

where \( T \) is guaranteed to be a monotonic decreasing function of \( |X_w| \).

For the second combination, consider that \( V(\Delta t) \) is a bell shape function which reaches its maximum when \( \Delta t = \tau \), the membrane potential could exceed the threshold only if the wavelet spectrum amplitude satisfies:

\[
|X_w| \geq X_{\text{th}} = \frac{u_{\text{th}} C_m}{e \tau^2}
\]  

(4.19)

in which case the firing delay \( T \) could be solved from:

\[
|X_w| = \frac{u_{\text{th}} C_m}{\tau T \exp \left( -\frac{T}{\tau} \right)}
\]  

(4.20)

Note that \( T \) is always less than \( \tau \) in (4.20), which ensures that \( T \) is a monotonic decreasing function of \( |X_w| \) when the amplitude spectrum \( |X_w| \) is larger than the threshold \( X_{\text{th}} \). If the wavelet spectrum amplitude is smaller than \( X_{\text{th}} \), the LIF neuron \( N_{\text{enc}} \) will not fire during \([t_{i}^{\text{enc}}, t_{i+1}^{\text{enc}}]\).
In both combinations discussed above, the wavelet spectrum of input signal $I_e$ is encoded into delay phase $T$ which is the difference between the time of each output fire and the arrival time of the most recent input spike in $C_{enc}$. Thus, larger wavelet spectrum amplitude corresponds to faster firing after each clock spike.

### 4.2.1 Encoding Neuron Implementation

Synapses and neurons as described in (4.7) through (4.11) are implemented in NEST with a single customized neuron model referred to as the Wavelet Sensor Neuron (WSN). In order to balance the accuracy and efficiency while solving ODEs for WSN, exponential integration method has been adopted to solve the state variable $u$, and Simpson’s rule was applied to the integration for state variable $v$:

$$u_{n+1} = P_{32}s_n v_n + P_{33}u_n \quad (4.21)$$

$$v_{n+1} = \frac{h}{6} \left( I_m(t_n) + 4I_m\left(t_n + \frac{h}{2}\right) + I_m\left(t_n + h\right) \right) + v_n \quad (4.22)$$

$$s_{n+1} = P_{33}s_n \quad (4.23)$$

$$I_m(t) = P_2 \left( 1 - \frac{\delta t^2}{\sigma^2} \right) \exp \left( - \frac{\delta t^2}{2\sigma^2} \right) I_e(t) \quad (4.24)$$

$$\delta t = t - T_i / 2 - t_{int} \quad (4.25)$$

where subscript $n$ indicates the $n$-th simulation step, $h$ is the simulation step size, $t_{int}$ is the arrival time of the most recent spike in $C_{int}$, and $P_2$, $P_{32}$, and $P_{33}$ are constant parameters defined by the following relations:
\[ P_2 = \frac{2}{\sqrt{3a\pi}^{1/4}} \]

\[ P_{32} = \frac{\tau}{C_m} \left[ 1 - \exp\left(-\frac{h}{\tau}\right) \right] \]

\[ P_{33} = \exp\left(-\frac{h}{\tau}\right) \]  

(4.26)

The WSN model incorporates two types of spike receptors to distinguish whether a spike is send to \( S_{\text{int}} \) or \( S_{\text{enc}} \), in the same manner as any other neuron model implemented in NEST which could receive spike input from more than one type of synapses. Input spikes with receptor type I are recognized as spikes sent to \( S_{\text{int}} \), which could reset \( v_0 \) to zero and set \( t_{\text{int}} \) to the current time; while input spikes with receptor type II are recognized as spikes sent to \( S_{\text{enc}} \), which in turn could reset \( u \) to \( u_c \) and \( s \) to zero.

A normal LIF neuron \( N_{\text{clk}} \) with an exponential decay synapse is implemented in this network as the clock generator. This LIF neuron is connected to itself with axon delay \( T_{\text{clk}} \) and synaptic efficacy large enough to generate a new output spike from itself. A short strong pulse injected to \( N_{\text{clk}} \) could initialize the first firing of \( N_{\text{clk}} \), and generate oscillatory clock spikes at constant interval approximate to \( T_{\text{clk}} \). These clock spikes are sent to type I receptors of WSN neurons with a short delay \( D_0 \), and type II receptors with a longer delay \( T_e \).

### 4.2.2 Encoding Human Voice Record

An encoding network was implemented to convert the human voice records obtained from Census Database of Carnegie Mellon University [86] (AN4) into spike trains related to the wavelet spectrum. An array of 100 WSNs with \( \tau = 45 \text{ ms} \) and \( \sigma \) varies between
0.2 ms and 10.0 ms were implemented in the encoding network. The spike trains could encode frequency components ranging from 100 Hz to 50 kHz in the input signal, which is wider than the human voice frequency limitations. Time constants $T_{\text{clk}} = 100$ ms, $D_0 = 1.0$ ms, $T_i = 45$ ms, and $T_e = 50$ ms was selected to meet all the requirements posed by (4.14). A negative threshold $V_{\text{th}} = -1.0$ mV was used in this implementation. The reset membrane voltage was set to $u_c = -2.72$ mV so that the longest spike delay is $T_{\text{max}} = 45$ ms, according to the solution of (4.15) with $V(T) = 0$ mV and $u(T_{\text{max}}) = u_{\text{th}}$. Since $T_e + T_{\text{max}} < T_{\text{clk}}$, there is always one output spike from each WSN within one clock cycle.

4.3 Results and discussion

The record file “an253-fash-b.raw” from the training set of AN4 database was used as the input to the WSN encoding network. The state variables of each WSN neurons were recorded for the testing purpose. A portion of the recorded variables of one WSN with $\sigma = 5.64$ ms was captured and plotted in Figure 4.2. Vertical red dash lines in Figure 4.2 represent the arrival times of the clock spikes for the type I synapse receptor of this neuron. Input $I_e$ was modulated with the wavelet kernel for 45 ms after each clock signal. When $I_e$ contains components matching the 5.64 ms time scale of the wavelet function, the WSN generates a larger modulated current, yielding as a consequence a larger state variable $v$. The clock spikes arrive at the type II synapse receptor of this WSN after 50 ms delay (indicated by the green vertical lines in Figure 4.2,
which at almost the end of each encoding period when \( I_e \) contains only higher frequency components (i.e., periods from 3120 ms to 3320 ms).

The voice record used in this experiment was the sound of female pronouncing the word “GO”. The output spikes of all 100 WSNs were raster-plotted for the time range from 3500 ms to 5000 ms using short vertical green bars as shown in Figure 4.3.
Continuous wavelet transform using Mexican-hat wavelet was also applied to the same voice record. The wavelet transform at translations $t_{\text{clk}} + 22.5$ ms were color coded and superimposed on Figure 4.3, where $t_{\text{clk}}$ are the firing times of $N_{\text{clk}}$. It could be found from Figure 4.3 that, the change of the fundamental frequency when pronouncing the word “GO” was clearly captured by the Mexican-hat wavelet transform, and the delay phases of WSN output fires were a good representation of the wavelet spectrum amplitudes during each clock cycle.

Such phase encoded spike trains are applicable to any supervised spiking neural learning methods such as ReSuMe method [9], Multi-spiking neural network [42], Synaptic Weight Association Training [8], Spike Pattern Association Neuron [10], or online learning with adaptive structure [87]; or any unsupervised learning methods such as Rank...
Order Learning [62] or Spiking Self-Organizing Map [66], [88]. Thus, the clustered or classified features of the frequency changes could be used to recognize the word pronounced. The phase delays of the WSN array in this example could substitute for the spectrogram in estimating key characteristics in speech recognition [89], and could support the building of speech perception system using ASNN.

4.3.1 Encoding Nonlinearity

The logarithm relationship between stimulation intensity and the delay phase of encoded spikes in sensory neurons was identified by many neurologists [90]. In many spiking neural network applications which implements PE as the sensing method, a log function was applied to the input signals to mimic the logarithm relationship [58], [65]. The WSN encoding method is highly nonlinear according to (4.18) and (4.20), yet the logarithmic relationship between stimulation intensity and the delay phase of spikes is a natural feature of the WSN encoding.

As shown in Figure 4.4, the linearity between $\log([^wX_w])$ and $\log(T)$ could be found in certain regions for the five selected WSN neurons with time constants $\tau$ being 20 ms, 40 ms, 60 ms, 80 ms and 100 ms, respectively. In Figure 4.4(a), positive firing threshold was adopted for these neurons, and the wavelet spectrum amplitude threshold was set to $X_{th} = 10^{-3}$ for all five neurons. The firing threshold $u_{th}$ for these neurons could be calculated by (4.19). We could find that WSN could encode $\log([^wX_w])$ to $\log(T)$ in a linear way when $|^wX_w|$ is in the linear region shown in Figure 4.4(a). Different time
constants \( \tau \) introduce different offsets to the linear relationship along the \( y \)-axis: larger \( \tau \) values corresponds to better encoding resolution for small \( |X_w| \).

As a comparison, negative firing threshold were used for the WSN neurons in Figure 4.4(b), with \( u_{th} \) all set to \(-0.2 \text{ mV}\). \( u_c \) for these neurons was adjusted according to:
such that the maximum output fire delay was always $T_{\text{max}} = 100$ ms. Linearity could also be found in the linear region indicated in Figure 4.4(b), when $T$ is a bit smaller than $T_{\text{max}}$. Different time constants $\tau$ introduce different offsets to the linear relationship along the $x$-axis.

It should be noted that, using the same $\tau$ settings, negative firing thresholds provide better logarithm linearity than positive firing thresholds for the encoding of signals with a larger range of $|X_w|$. Since the parameter $\tau$ in the WSN neuron is limited by the encoding window length, negative firing thresholds could be a better choice when the encoding linearity is of interest, as demonstrated in this paper when encoding was performed on the example of the human voice record. However, the $|X_w|$ cut-off feature provided by the positive firing thresholds could be useful when only large values of $|X_w|$ are of interest, such as in the case of IS detection for EEG recordings. The threshold configuration as well as the time constant $\tau$ should thus be carefully selected for a given application, so that the features of interest in the input signal could be best encoded into the delay of output fires.

### 4.3.2 Kernel Function Variation

Although a shifted Mexican-hat wavelet mother function was used for the post-synaptic current shape function in $S_{\text{int}}$, it is not required for the WSN neuron to work properly. Any types of wavelet mother functions could be used as the current shape function in
WSN, and the input signal will be decomposed according to the mother wavelet functions selected. If a discrete wavelet is demanded, the kernel function of the discrete wavelet at different time scale with proper shifting should be used as $\Psi$ in (4.10).

More interestingly, since the integration of the wavelet kernel performs only in a limited time duration, the only requirement for $\Psi(t)$ is that:

$$\lim_{t \to \pm \infty} \Psi(t) = 0$$

(4.28)

and $\Psi(t)$ is not required to be absolutely integrable and square integrable from $-\infty$ to $+\infty$.

Some functions, such as the alpha function:

$$\Psi_\alpha(t, \sigma) = \frac{t}{\sigma} \exp\left(-\frac{t}{\sigma}\right)$$

(4.29)

could also be used to decompose the input signals.
Figure 4.5 Comparison between WSN encoding using Mexican-hat wavelet and Alpha function. The $s_m$ are the time scales used for WSNs with Mexican-hat wavelet kernels while $s_\alpha$ are the corresponding time scales used for WSN with Alpha function kernels.
The results shown in Figure 4.5 illustrate the encoding of Human Voice Record using Mexican-hat function and Alpha function as the decomposing kernels. The phase encoding using WSNs with Mexican-hat kernel yields similar results compared to those with Alpha function kernel. Thus, the time scales of Mexican-hat kernel are related to those of Alpha functions with the following constant ratio:

$$\frac{\sigma_n}{\sigma_\alpha} = 2.784 .$$  \hspace{1cm} \text{(4.30)}

This is a reasonable representation of the differences between the central frequencies of the Mexican-hat kernel and the Alpha function kernel. Functions, such as the Alpha function illustrated above, are more biologically plausible, thus more likely to be implemented for WSN encoding on an analog ASNN simulation platform.

### 4.3.3 Temporal Resolution

Since the wavelet of a WSN is convolved with the input signal only once during each clock cycle, the encoding temporal resolution of one WSN is limited to the clock interval $T_{\text{clk}}$. Considering that the total of integration time $T_i$ and that encoding time $T_e$ should be smaller than $T_{\text{clk}}$, and the time constant $\tau$ should also be smaller than $T_i$, although a decreased $T_{\text{clk}}$ could enhance the encoding temporal resolution, it might also harm the encoding range of the wavelet spectrum amplitude. In order to enhance the temporal resolution of a WSN array without interfering with the encoding range, we could still implement multiple WSNs for each time scale selection, but with different $D_0$ values. Accordingly, the wavelet transform would be performed at different translations within
each clock cycle, and thus could significantly enhance the temporal resolution of the
coding without shrinking the length of each clock cycle.
5. INTER-ICTAL SPIKE DETECTION

5.1 Neuron Models and Network Structure

The WSN model introduced in section 4.2 was adopted as the encoding neuron, with Mexican-hat wavelet chose as the encoding kernel function. As shown in Figure 5.1, a single regular spike LIF neuron \( N_{\text{clk}} \) is implemented as the clock neuron, which is recursively stimulated by its own output. An initializing stimulation \( I_{\text{init}} \) is designed as a short pulse sufficient to initial the first output spike in \( N_{\text{clk}} \), the output spikes from \( N_{\text{clk}} \) feed back to the clock neuron itself with a time delay \( T_{\text{clk}} \), and a synapse weight sufficient to induce another output spike from \( N_{\text{clk}} \).

This configuration ensures that the clock neuron could generate series of output spikes with intervals approximate to \( T_{\text{clk}} \). These output clock spikes are sent to the two synaptic channels of all WSNs, with two different delays indicated by the green color and blue color in Figure 5.1. The WSNs are grouped according to the EEG channels. The EEG amplitudes in each channel are preprocessed and shared by all WSNs in one group, and the output from WSNs are organized are grouped by EEG channels, and sorted according to their corresponding time scales, which is ready for further processing in other ASNN structures.
Figure 5.1 Network structure to detect interictal spikes from EEG records
5.2 EEG Records and Inter-ictal Spike Detection

The features of interests in the EEG recordings are the IS as described in [74], which in epilepsy are a key feature used for 3D source localization of seizure onsets. The detection of IS will also help delineate EEG records that could lead to seizures [91]. IS could be found synchrony in multiple channels between ictal events, characterized as fast EEG transients (faster than 50 ms) with steep rising and falling slopes, and habitually followed by a slow potential. Since the shape of IS is similar to the Mexican-hat wavelet mother function, WSN with time scale matched to the duration of these spikes will generate much faster output spikes. We built 50 WSNs for each EEG channel, with $\sigma$ varies between 5 ms and 70 ms. A positive threshold $u_{th} = 0.5$ mV was implemented in this network, which ensures that WSNs only fire when IS is detected, and remain quiet otherwise. All channels of EEG recordings are pre-processed by a fifth-order Butterworth high-pass filter to avoid the influence of stochastic drifts. A 5 Hz cut-off frequency was used for this high-pass filter, so that high frequency features, especially the IS peaks are well preserved after the pre-processing.

Scalp EEG recordings from a patient with focal epileptic seizures was used as input signal in this application. The EEG recordings consist of 19 individual channels, each sampled at 512 Hz frequency. We implemented 19 groups of WSN arrays, each connected to one individual channel of the EEG signal. Time constants $T_{\text{clk}} = 200$ ms,
\( D_0 = 1.0 \text{ ms}, \; T_i = 85 \text{ ms}, \; T_c = 100 \text{ ms} \) and \( \tau = 100 \text{ ms} \) were used for all WSNs in this network.

### 5.3 Results and Discussion

A portion of multi-channel EEG recordings with IS identified was plotted in Figure 5.2, as well as the WSN encoding results. Note that the delays of output spikes are color coded for each clock cycle and those clock cycles with no output spike were filled with “NAN” values and plotted as white blocks.

We could find from Figure 5.2(a) that there are two occurrences of synchronized IS in this segment of EEG recording. The zoomed view of the first IS at 100 ms scale as illustrated in Figure 5.2(c) show that identified interictal spikes meet the criteria that characterize them, including sharp rising and falling edges, larger peak amplitudes and synchronization in multiple channels. These two IS strikes were accurately captured by the WSN encoding network. In Figure 5.2(c), it was determined that only the WSN groups which were connected to the EEG channels with IS presented fired output spikes, and the WSN with wavelet time scale best matches the IS width fires faster than other WSNs in the same group. Although there were false positive WSN output spikes generated at occasions when no IS was identified, the temporal and spatial sparseness of these output spikes make it possible to filter out such false positives using any spiking learning methods.
Figure 5.2 Encoding of interictal spikes in a segment of multi-channel EEG recordings: (a) the waterfall plot of the recorded EEG potentials; (b) the pseudo color mapping of the encoded output spike delays; (c) detailed view of the IS wave found at approximately 18th second in the upper portion, and output spikes generated by WSNs in the lower portion.
6. CONCLUSIONS AND FUTURE WORK

Clearly the GLIF model introduced in this study could accurately simulate the dynamics of a neuron cell membrane potential if the parameters and kernel function were extracted and fitted meticulously using the statistical method we provided. Although the reference model we used to demonstrate the extracting procedure and to test the GLIF model was the HH model with two ionic channels, our GLIF model could be easily fitted to any complicated HH model with more ion channels. This last assertion can be supported simply by plotting the trace of the conductance after each spike for each ion channel and fit them to the kernel function.

The comparison of the GLIF model performance with that of the NLIF shows that our model provides much better calculation accuracy in simulating the biological neuron activity. After using the GEMA method for the GLIF model, the calculation complexity was kept to an acceptable level. Such outcomes increase the prospects of the GLIF model for its implementation in larger scale and real-time ASNNs. As we seek to reach this implementation goal, future research work will focus on building more biologically plausible ASNN using the GLIF neuron model on parallel computation platforms such as the General Purpose Graphic Process Unit (GPGPU) and Field-Programmable Gate Array (FPGA) devices, and applying such ASNN to resolve a multitude of real-world problems associated with pattern recognition and pattern classification, among other things.
Encoding of analog signals into spike trains is one of the most important steps for information processing in biological nervous systems. The encoding method we proposed in this paper incorporates the concepts of synaptic current modulation with phase encoding representation. We proved that the proposed WSN model combining a preprocessing unit and a LIF neuron could perform the wavelet decomposition of the input signal, and convert the wavelet spectrum amplitude at certain translation and time scales into the output fire delay of the WSN neuron.

Encoding networks using WSN neurons were implemented in this study to encode an example of a human voice record, with results that are quite similar to continuous wavelet decomposition. The encoding method was also applied to multi-channel EEG records of epilepsy patients to detect the IS to guide the prognosis of epilepsy and to determine the 3D source localization of the epileptogenic zone. We found that when the WSNs contains wavelet kernel with time scales matched to the expected durations of IS, they could detect these IS events, and accurately preserve the synchronizing behavior of IS in the output spike trains for enhanced diagnosis and 3D source localization. The linearity property and limitations of mother wavelet functions of this WSN encoding method were discussed as a guidance for choosing proper parameters for the WSN network to fit a specific application. We also provide a simple method to overcome the temporal resolution limitation posed by the clock signal, so that the wavelet decomposition could be performed with higher temporal accuracy if needed.
Beyond the above contributions, this work also provides an intuitive insight of how stimulations gathered by sensor neurons might be represented and processed by a biological nervous system: the modulation behavior found between dendrites together with the integration feature of a biological neuron could perform decomposition of stimulation signals similar to wavelet transforms, and encode only those features of interest in the stimulation into the spike delay phases.

There are other possibilities for using the proposed encoding method such as: (1) apply graph theory [92] to find the connectivity between encoded spike trains, or (2) build spiking self-organizing map and supervised learning systems to further process the encoded spike trains, and classify the patterns represented by the encoded spike trains into meaningful symbols. Since wavelet decomposition is also an important tool in feature extraction step for EEG signal processing [93], WSN neurons could facilitate EEG processing with a new online and biologically plausible way to extract features from the EEG channels. EEG classification tasks based on certain frequency bands [94], [95] could also benefit from the multidimensional transform inherited in the WSN encoding method. Although the WSN and the encoding network was implemented in the NEST environment, which is based on a digital computing platform, the concepts of WSN is fully compatible with analog computing. We are interested in developing analog circuits to implement WSN encoding network, so that Ultra Large Scale Integration methods could be used to build a highly parallel neuromorphic system.
LIST OF REFERENCES


APPENDICES

Appendix A. Solving for the GLIF ODE

The first order ordinary differential equation in (3.1) could be solved analytically. Regardless of any changes of $t^f$, we could always define

$$P(t) \equiv g_{\text{kern}} (t - t^f)/C_m$$
$$Q(t) \equiv \left[ i_s(t) + i_{\text{kern}} (t - t^f) \right]/C_m,$$

and a general solution of (3.1) could be found as

$$u(t) = \left[ \int Q(t)e^{\int_{t^f}^{t} P(\tau)d\tau} d\tau + C \right] e^{\int_{t^f}^{t} P(\tau)d\tau}. \quad (A.2)$$

Consider the closest firing time $t^f$ before $t$, and note that $u(t^f) = u_{th}$, the constant $C$ could be solved as:

$$C = u_{th} e^{\int_{t^f}^{t} P(\tau)d\tau} - \int_{-\infty}^{t^f} Q(\tau)e^{\int_{\tau}^{t} P(\tau)d\tau} d\tau. \quad (A.3)$$

Hence the particular solution of (3.1) is

$$u(t) = \left[ \int_{-\infty}^{t^f} \frac{Q(\tau)}{\kappa(\tau)} d\tau + \frac{u_{th}}{\kappa(t^f)} \right]$$
$$\left. -\int_{t^f}^{t} \frac{Q(\tau)}{\kappa(\tau)} d\tau \right] \kappa(t)$$
$$= \kappa(t) \left[ u_{th} + \int_{t^f}^{t} \frac{Q(\tau)}{\kappa(\tau)} d\tau \right], \quad (A.4)$$

where
Considering that survival time \( t_r \equiv t - t_f \) for any \( t \) and \( t_f \), the solution can be derived as

\[
\kappa(\tau) = e^{-\int_{-\infty}^{\tau} p(\xi) d\xi} \quad (A.5)
\]

\[
\text{Since the term}
\]
\[
e^{-\int_{-\infty}^{t_f} p(\xi) d\xi}, \quad (A.7)
\]

\[
\text{is not related to the integral variable } \tau, \text{ it could be moved out of the integration to yield}
\]
\[
u(t) = u_m e^{-\int_{0}^{t_f} p(\xi + t_f') d\xi} + e^{-\int_{0}^{t_f} p(\xi) d\xi} \int_{0}^{t_f} Q(\tau + t_f') e^{-\int_{\tau}^{\tau + t_f'} p(\xi) d\xi} d\tau \quad (A.6)
\]
\[
= u_m e^{-\int_{0}^{t_f} p(\xi + t_f') d\xi} + e^{-\int_{0}^{t_f} p(\xi) d\xi} \int_{0}^{t_f} Q(\tau + t_f') e^{-\int_{\tau}^{\tau + t_f'} p(\xi) d\xi} d\tau.
\]

\[
\text{If we define}
\]
\[
\Pi(t_r) = e^{-\int_{0}^{t_f} g_{km}(\xi) / C_n d\xi}, \quad (A.9)
\]

the solution could be rewritten as
\[ u(t) = u_{th} \Pi(t_r) + \Pi(t_r) \int_0^{t_r} \frac{i_{kern}(\tau)}{C_m \Pi(\tau)} d\tau + \Pi(t_r) \int_0^{t_r} \frac{i_s(\tau + t^f)}{C_m \Pi(\tau)} d\tau. \]  

(A.10)

The first two terms of this solution could be defined as:

\[ u_{kern}(t_r) = \Pi(t_r) \left[ u_{th} + \int_0^{t_r} \frac{i_{kern}(\tau)}{C_m \Pi(\tau)} d\tau \right], \]  

(A.11)

which is a function describes the post-fire membrane voltage trajectory at no input spikes.

The influence of input spikes is described by the third term in (A.10), which could be defined as:

\[ u_{syn}(t_r) = \Pi(t_r) \int_0^{t_r} \frac{i_s(\tau + t^f)}{C_m \Pi(\tau)} d\tau. \]  

(A.12)
VITA

ZHENZHONG WANG

1984 Born, Nanjing, China

2006 B.S., Applied Physics
Southeast University
Nanjing, China

2009 M.S., Microelectronics and Solid-state Electronics
Southeast University
Nanjing, China

2015 Ph.D. candidate, Electrical Engineering
Florida International University
Miami, Florida

PUBLICATION AND PRESENTATIONS


