Model and Simulation of Izhikevich Phasic Spiking Neuron

Samuel J. Wenke WENKESJ@MAIL.UC.EDU
EECS CEAS
University of Cincinnati
2600 Clifton Ave.
Cincinnati, OH 45221, USA

Abstract

This document describes the modeling and simulation of Izhikevich neuron. In particular, these experiments focus on the implementation of the Phasic Spiking Neuron. A simulation enviorment was created to explore and observe the models behaviors at varying scales. The first experiement consisted of implementing a program to simulate the neuron and drive it with different levels of external input. The second experiment simulated a network of 2 neurons from the before mentioned implementation. Plots for both of these experiments were obtained and results are described. The simulation environment was designed for discrete parralel execution of multiple neurons and multiple connections.

1 Introduction

One of the most widely used current models for neurons is the Izhikevich model, proposed by Eugene Izhikevich (Izhikevich, 2003, 2004). It models the membrane potential of a neuron by two coupled differential equations:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I(t) \tag{1}$$

$$\frac{du}{dt} = a(bv - u) \tag{2}$$

with the condition that:

$$if\ \ v \geq 30, \quad v = c, \ \ u = u + d \tag{3}$$

Thus, the model is defined completely by four parameters: *a, b, c* and *d*. By setting the parameters to different values, neurons with many different natural behaviors can be obtained, as described in Izhikevich (2003, 2004). The model generates the full shape of the action potential. Thus, it is more than an integrate-and-fire model but far less complicated than a Hodgkin-Huxley model. The experiment is divided into 3 sections.

The first section documents the implementation of a single Phasic Spiking Neuron and gives plots and draws conclusions from them.

The second section documents the implementation of a network of 2 Phasic Spiking Neurons; which a neuron *A* feeds its output into a neuron *B*. Results were plotted and conclusion were drawn for each neuron.

The third section explores a broader, parallel approach to the Izhikevich neuron simulation and provides documentation for supplemental experiments performed.

2 Implementation of a Single Neuron

Serveral simulations of a single Phasic Spiking Neuron were performed with varying inputs *I.* A discrete time series was created to simulate a real time implemention. The time series begins at $t = 0$, on an interval of each step $t = t + 0.25$ and ending at $t = 500$. At a specified time *T1*, the neuron will begin receiving the input *I.* By default, the neuron will receive an input of 0 before time *T1*.

The output of a single neuron receiving steady input of $I = 1.0$ is shown in Figure 1. The spike rate over the last 300 time steps is shown in Figure 2. The output *[eq. 3]* of the neuron simulated with inputs I = 1, 5, 10, 15, 20 is shown in Figures 1, 3, 4, 5, 6.

2.1 Results

As an input *I* is supplied steadily over a discrete time interval span of *t*, the neuron will produce spikes at a steady rate proportional to the value of the input *I.* The mean spike rate of a neuron increases almost linearly as the input *I* increases as seen in Figure 2. The neuron is showing a functional relationship between the amount of spikes over time with the input *I*.

2.2 Plots

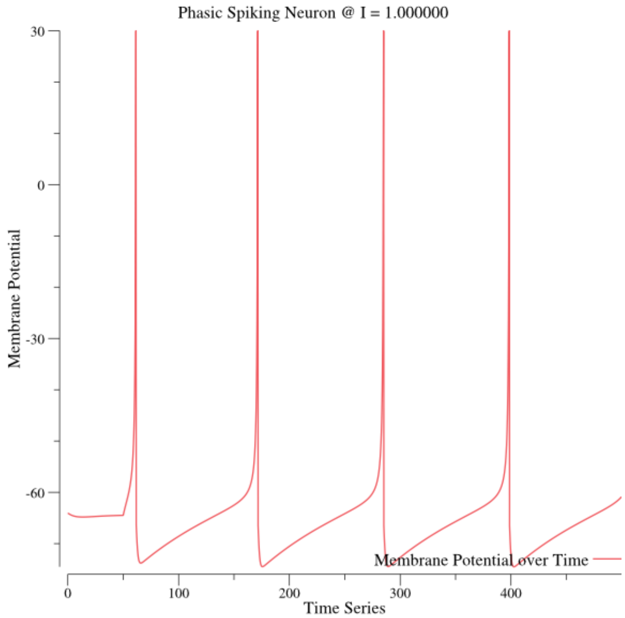

Figure 1. Single Phasic Spiking Neuron receiving steady Input I = 1.0

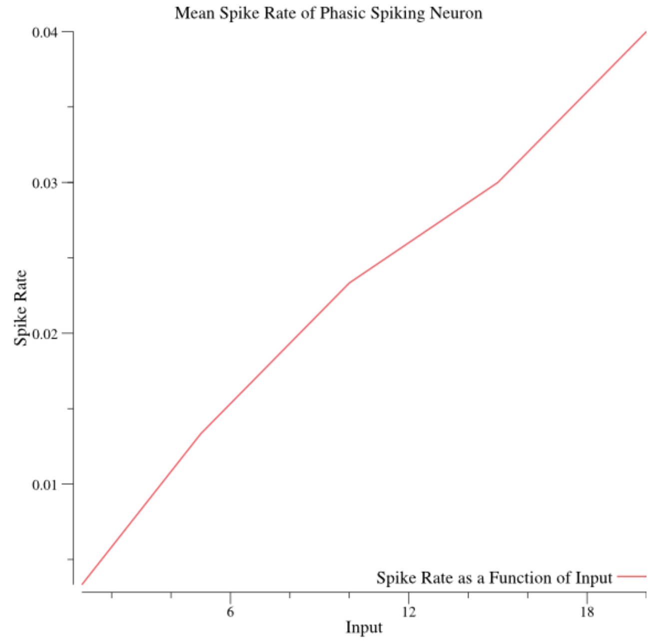

Figure 2. Mean Spike Rate of a single Phasic Spiking Neuron over input *I*.

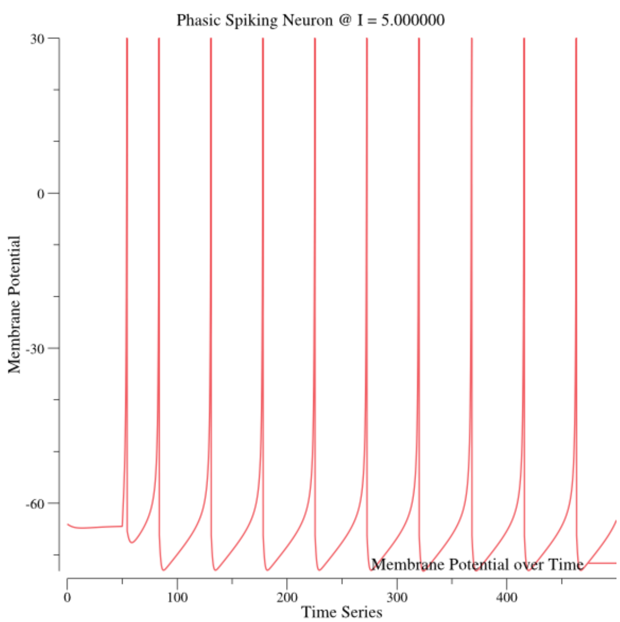

Figure 3. Single Phasic Spiking Neuron receiving steady Input I = 5.0

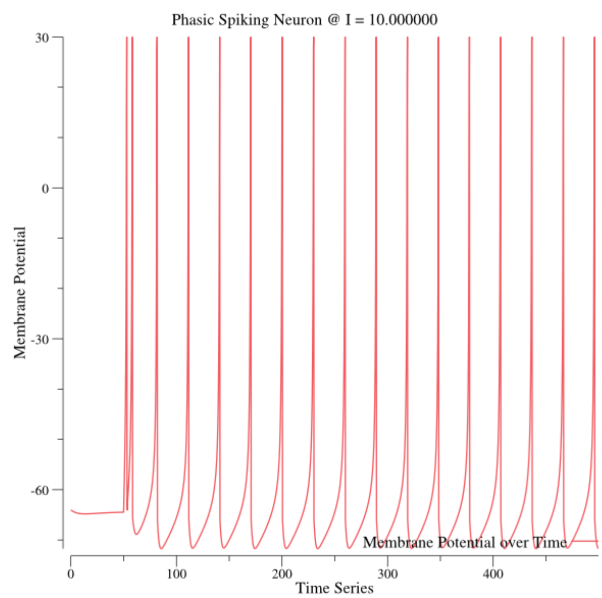

Figure 4. Single Phasic Spiking Neuron receiving steady Input I = 10.0

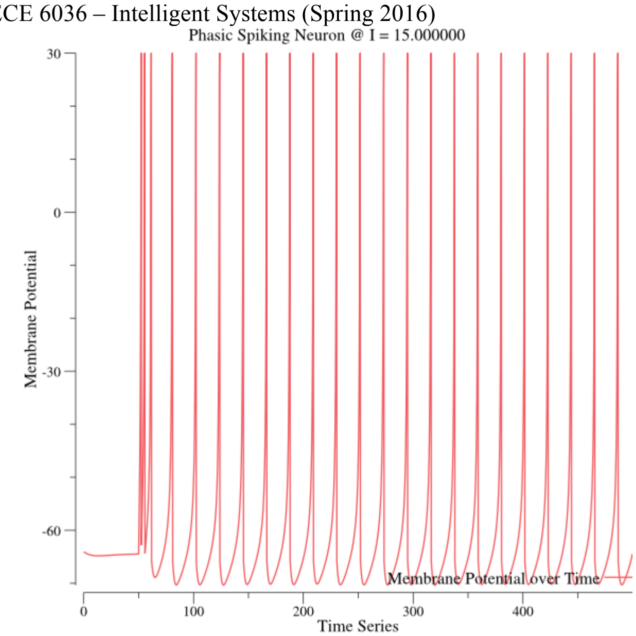

Figure 5. Single Phasic Spiking Neuron receiving steady Input I = 15.0

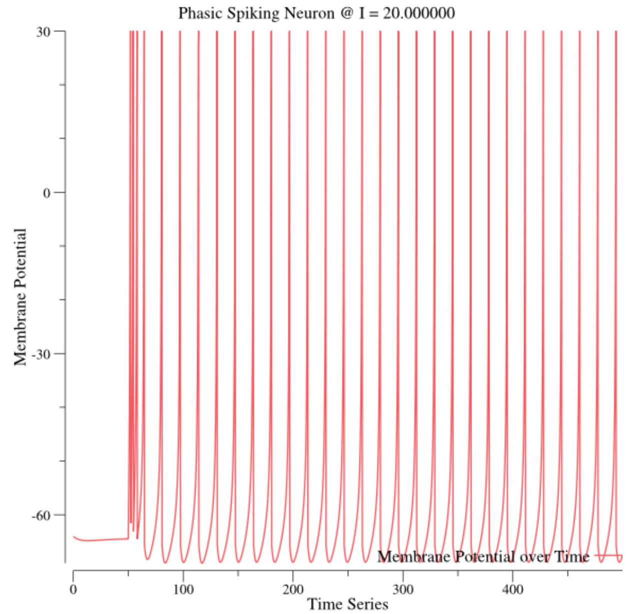

Figure 6. Single Phasic Spiking Neuron receiving steady Input I = 20.0

3 Implementation of a Simple Two Neuron Network

Serveral simulations of a simple Spiking Neuron network with 2 neurons were performed with varying inputs I_A, similar to the first experiment. For the network to produce significant results, a neuron, so named neuron B, was connected to another neuron, so named neuron A. The connection symbolizes the output of neuron A feeding into the input I_B *[eq. 4]* of neuron B. The equation for I_B is shown:

$$I_B(t) = w y_A(t) \tag{4}$$

This connection only supplies input to neuron B when the following conditions are met:

$$y_A(t) = 1 \quad if \;\; V_A(t) = 30, \quad else \;\; y_A(t) = 0 \tag{5}$$

Where V_A is the output of neuron A at the corresponding time t. The input of neuron B is calculated as the product of the weight w of the connection from neuron A to neuron B and the conditional output y_A of neuron A.

The same conditions were applied to neuron A and identical output results were obtained and plotted in Figures 7, 9, 10, 11, and 12. The mean spike rate of neuron A was indentical to the first experiment and is plotted in Figure 8. For neuron B, results are plotted in Figures 13, 15, 16, 17, and 18. The mean spike rate of neuron B was plotted in Figure 14.

3.1 Results

As input I is steady over a discrete time interval, neuron A will produce spikes at a steady rate proportional to the value of the input I similar to the first simulation. The mean spike rate of a neuron increases almost linearly as the input I increases as seen in Figure 8. In this simulation, neuron B doesn't receive steady input from any source; neuron A will only provide input to neuron B if V_A *[eq. 2]* is above the threshold $V_A = 30$. At time $T1$, neuron A and neuron B will produce spikes at the same intervals. As input I_A increases, neuron A have an increased spike rate which will in turn produce a steady output to neuron B. But on the contrary, neuron B will not produce a steady input I_B like neuron A. Neuron B is simulating a refatory period in between spikes from neuron A. It won't produce enough output V_B which depends on I_B at time $T1$ when neuron A fires. Neuron B will in turn have a lower mean spike rate over the discrete time series t with increasing steady input I_A into neuron A as seen in Figure 14 compared to Figure 8.

3.2 Plots

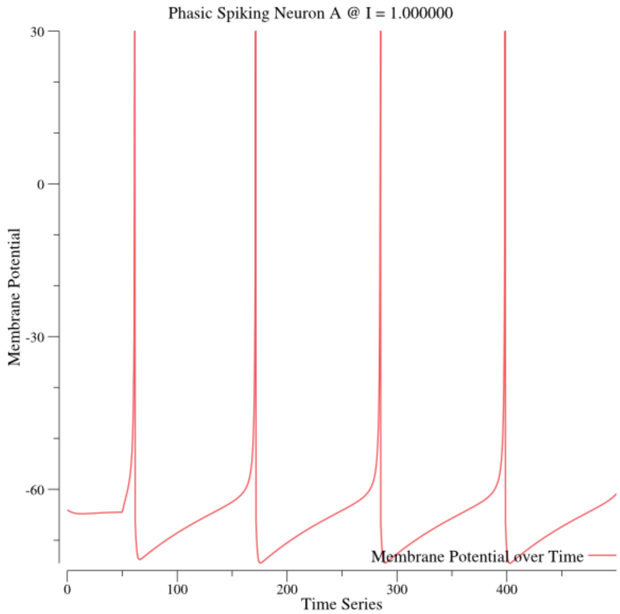

Figure 7. Phasic Spiking Neuron A receiving steady Input I = 1.0

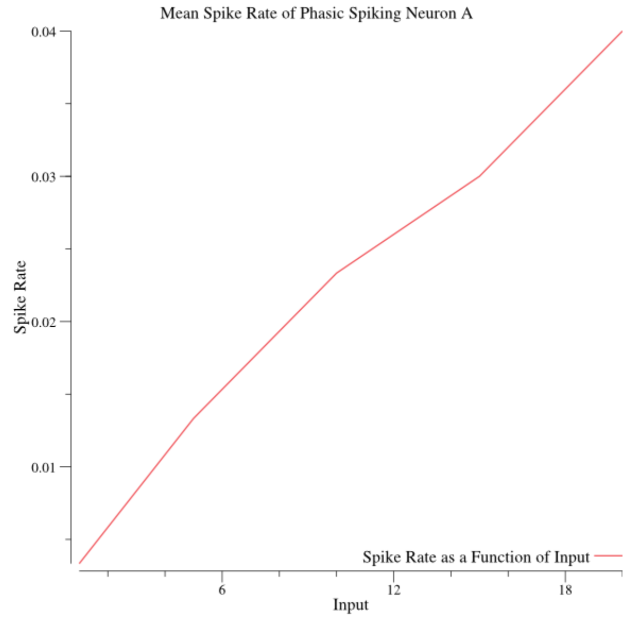

Figure 8. Mean Spike Rate of Phasic Spiking Neuron A over input I.

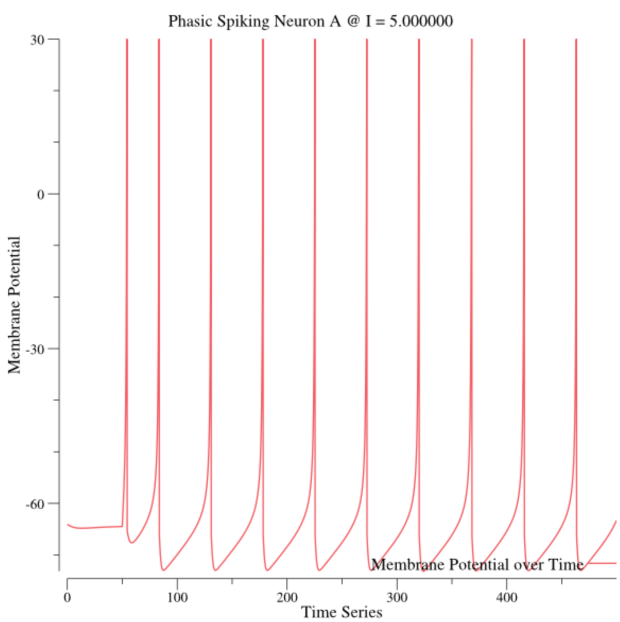

Figure 9. Phasic Spiking Neuron A receiving steady Input I = 5.0

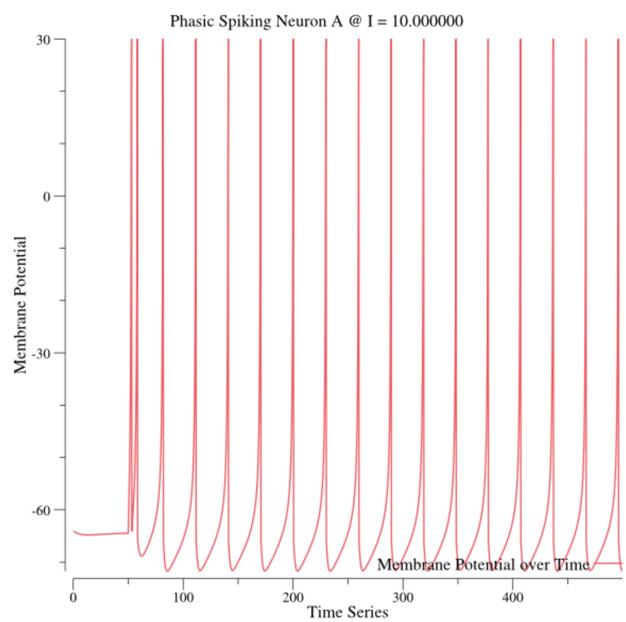

Figure 10. Phasic Spiking Neuron A receiving steady Input I = 10.0

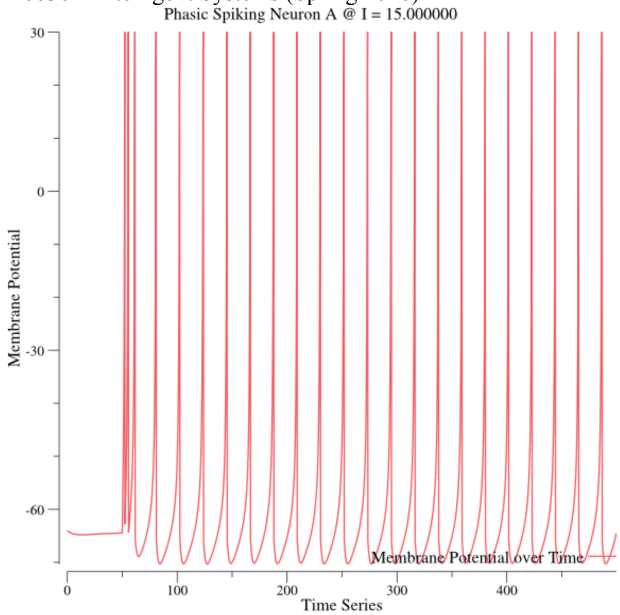

Figure 11. Phasic Spiking Neuron A receiving steady Input I = 15.0

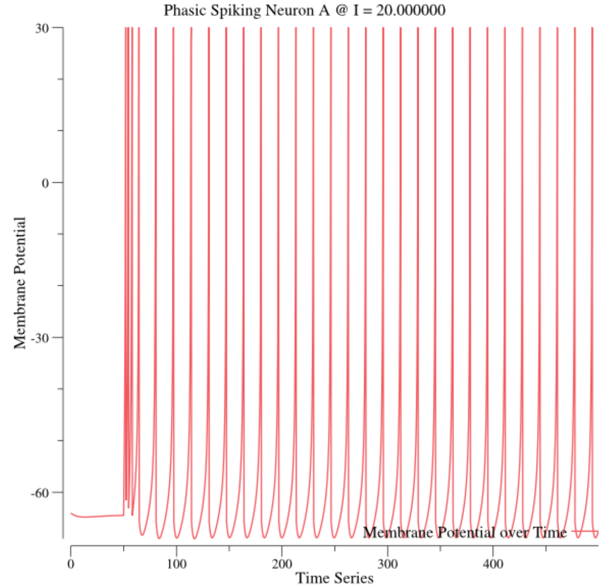

Figure 12. Phasic Spiking Neuron A receiving steady Input I = 20.0

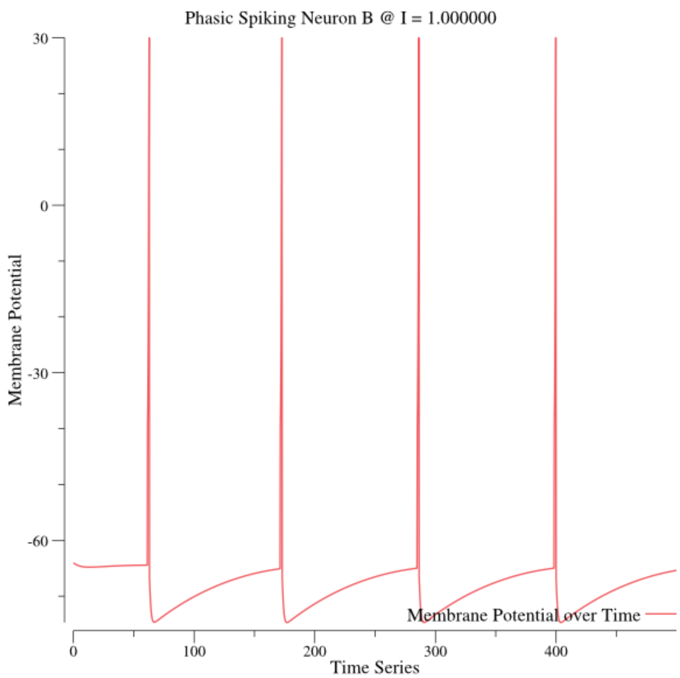

Figure 13. Phasic Spiking Neuron B receiving steady Input I = 1.0

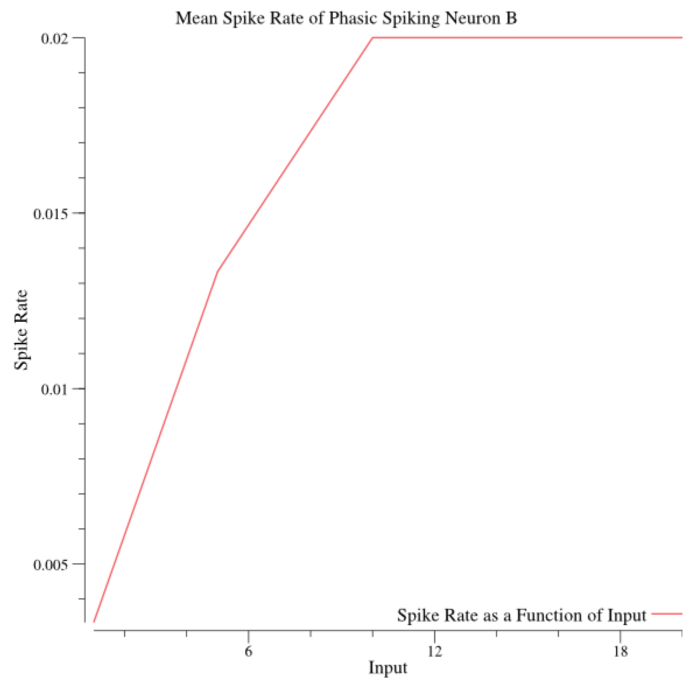

Figure 14. Mean Spike Rate of Phasic Spiking Neuron B over input I.

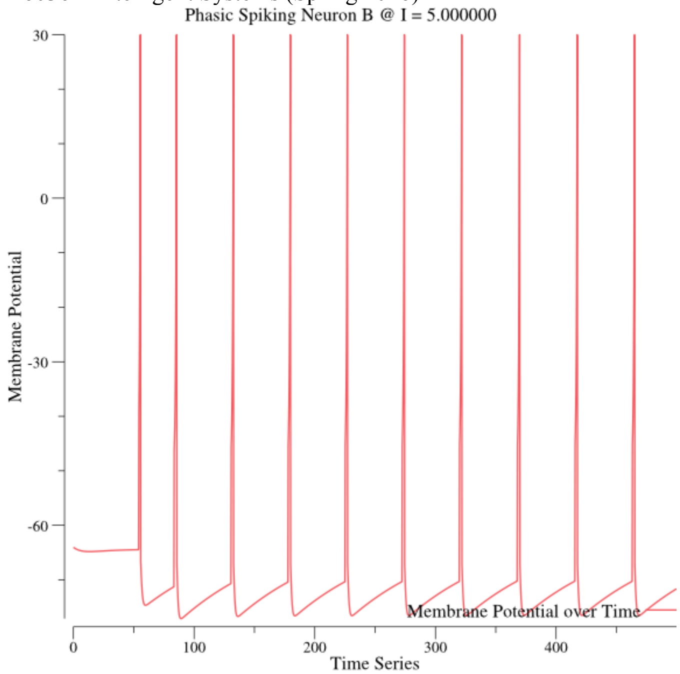

Figure 15. Phasic Spiking Neuron B receiving steady Input I = 5.0

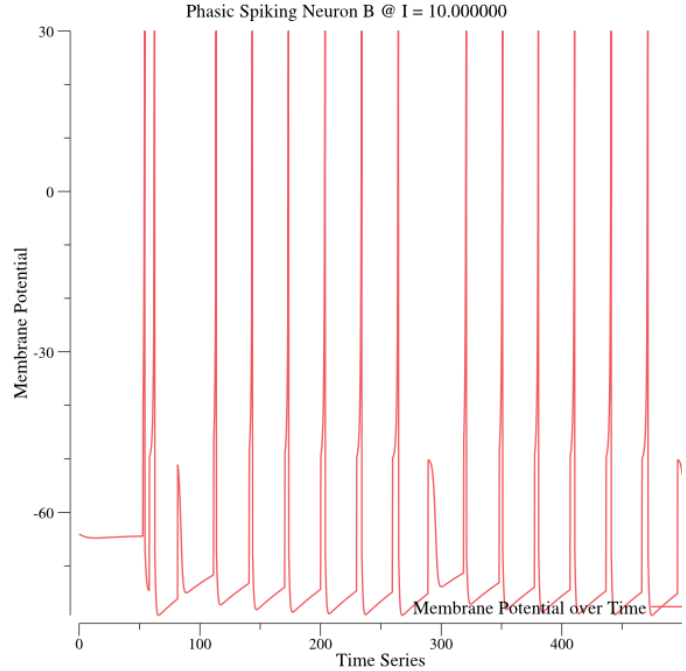

Figure 16. Phasic Spiking Neuron B receiving steady Input I = 10.0

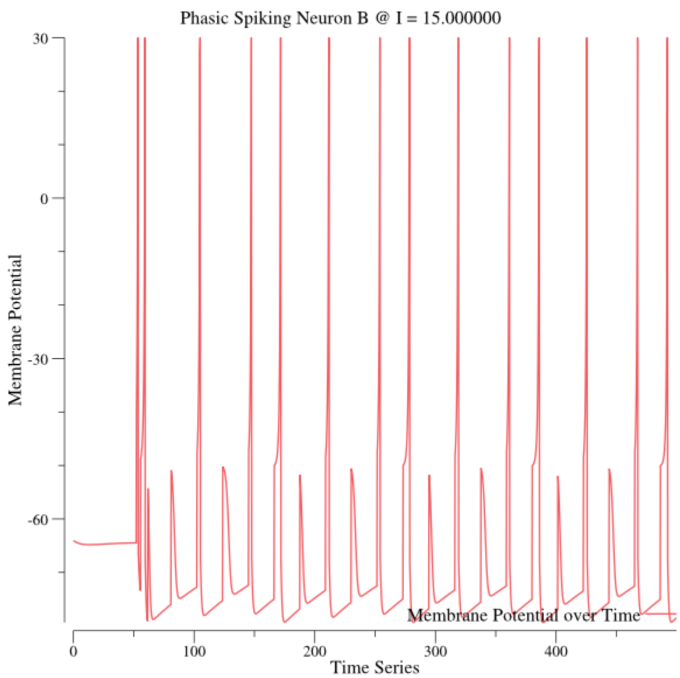

Figure 17. Phasic Spiking Neuron B receiving steady Input I = 15.0

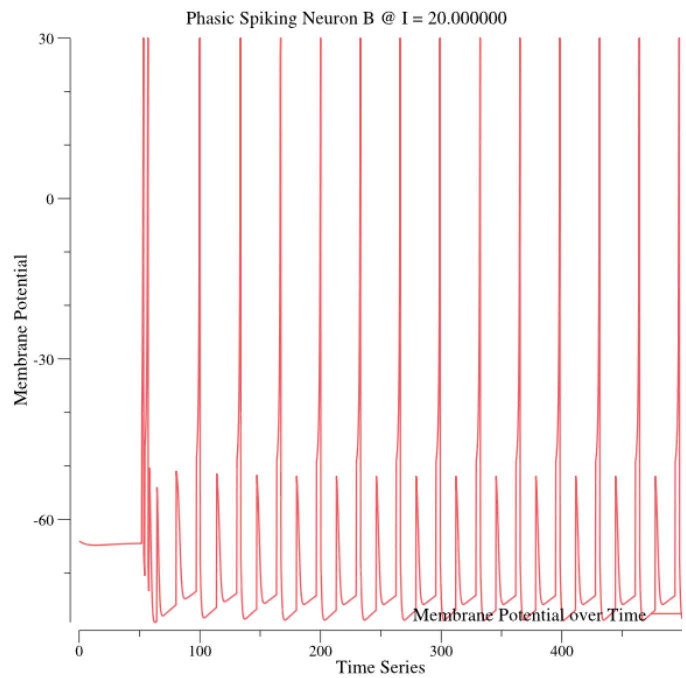

Figure 18. Phasic Spiking Neuron B receiving steady Input I = 20.0

4 Discussion

The set of simulation results supplied in this document are generated from code written by the author. The code is supplied under an MIT liscence. The simulation of the *Phasic Spiking Neuron Network* was successfully implemented by creating a concurrent discrete time series environment. Neuron N is represented by a single routine which simulates its own time series loop (on time t) and independentanly calculating V_N. Neuron N communicates over connections with weight w_M where M is the number of connections. A number of load/store operations and parallel routines are implemented to provide the needed functionality across connections. The simulations shown only involves 2 neurons, where as the simulation environment created can include N neurons. N can be upto*, but not limited to*, a combination of the number of concurrent operations and memory of the simulation computer or computer network. Latest code and information related to this project can be found at http://github.com/wenkesj/sn The experiment produced satisfying results and gave the author a better understanding of neural networks and discrete time series simulation.

Acknowledgements

Dr. Ali Minai, for the use of his introduction and inspiration in the document associated with this assignment.