Summer Fellowship Report

On

Scilab Case Studies

Submitted by

Sumat Dhuwariya
Department of Computer Science
VIT Bhopal University

Under the guidance of

Prof. Prabhu Ramchandran
Aerospace Engineering Department
IIT Bombay

Mentor

Mrs. Rashmi Patankar

July 1, 2026

Acknowledgment

I would like to express my deepest gratitude to the FOSSEE Team, IIT Bombay, for offering me the invaluable opportunity to participate in the FOSSEE Summer Fellowship 2026. I am immensely grateful to Prof. Prabhu Ramchandran for spearheading this initiative, which plays a pivotal role in promoting free and open-source software in education.

A special note of thanks goes to my mentor, Mrs. Rashmi Patankar, for her unwavering guidance, valuable feedback, and encouragement throughout the fellowship. Her expertise significantly enriched my learning journey and was instrumental in the successful completion of my projects.

I would also like to express my gratitude towards my family, friends and members of VIT Bhopal University, for their kind co-operation and encouragement, which helped me complete this Internship.

Sumat Dhuwariya
VIT Bhopal University

Contents

List of Figures

Chapter 1

Introduction

To understand how the brain works, computational neuroscientists must combine biological experiments with the numerical simulation of large-scale neural networks. However, developers and researchers constantly face a strict compromise between biological plausibility and computational efficiency. Biophysically detailed models, such as the Hodgkin-Huxley model, accurately reproduce neural firing patterns but are computationally prohibitive for large-scale network simulations. Conversely, simpler artificial models fail to capture the rich, diverse firing dynamics of real cortical neurons.

The Free and Open Source Software in Education (FOSSEE) initiative promotes the adoption of open-source tools to enhance technical education. Scilab is a prime example of a cross-platform numerical computation environment that provides a robust alternative to proprietary software. With its powerful mathematical engine, Xcos visual simulation environment, and interactive GUI capabilities, Scilab is highly capable of handling advanced neurocomputational physics.

During this fellowship, I successfully completed three progressive case studies bridging Neurobiology and Artificial Intelligence within Scilab:

- **Case Study 1:** Spiking Neural Networks for Handwritten Image Classification using Scilab.
- **Case Study 2:** Continuous-Time Simulation of Neural Synapses Using First-Order Kinetics in Xcos.
- **Case Study 3:** Simulating the Izhikevich Model, Cortical Rhythms, and GUI Dashboard Development in Scilab.

These studies demonstrate the versatility of Scilab in modelling biological systems, discrete-time Euler integration, and graphical interface deployment without relying on external machine learning or neuroscience toolboxes.

Chapter 2

Case Study 1: Spiking Neural Networks for Image Classification

2.1 Abstract

This project presents handwritten digit classification using a single-layer Spiking Neural Network (SNN), implemented from scratch within Scilab. An 8×8 image dataset (3823 images) was used with an 80-20 train-test split. Static pixels were translated into stochastic spike sequences using Poisson encoding, processed via the Leaky Integrate-and-Fire (LIF) model. A supervised Spike-Perceptron learning rule was deployed with momentum-assisted weight updates and L2 regularization. The model achieved a final, reproducible classification accuracy of 91.90%.

2.2 Problem Statement

While Artificial Neural Networks (ANNs) excel in cognitive tasks, they utilize continuous activation functions requiring significant computational resources. Spiking Neural Networks (SNNs), the third generation of neural networks, offer energy-efficient discrete spike processing. The challenge lies in mathematically translating static images into time-series spike data and updating synaptic weights to train the network without utilizing specialized Python machine learning libraries.

2.3 Methodology

- **Poisson Spike Encoding:** Pixels are converted into spikes based on normalized brightness probability over 100 ms simulation windows.
- **LIF Integration:** The discrete-time Euler integration solves the membrane leakage differential equation:

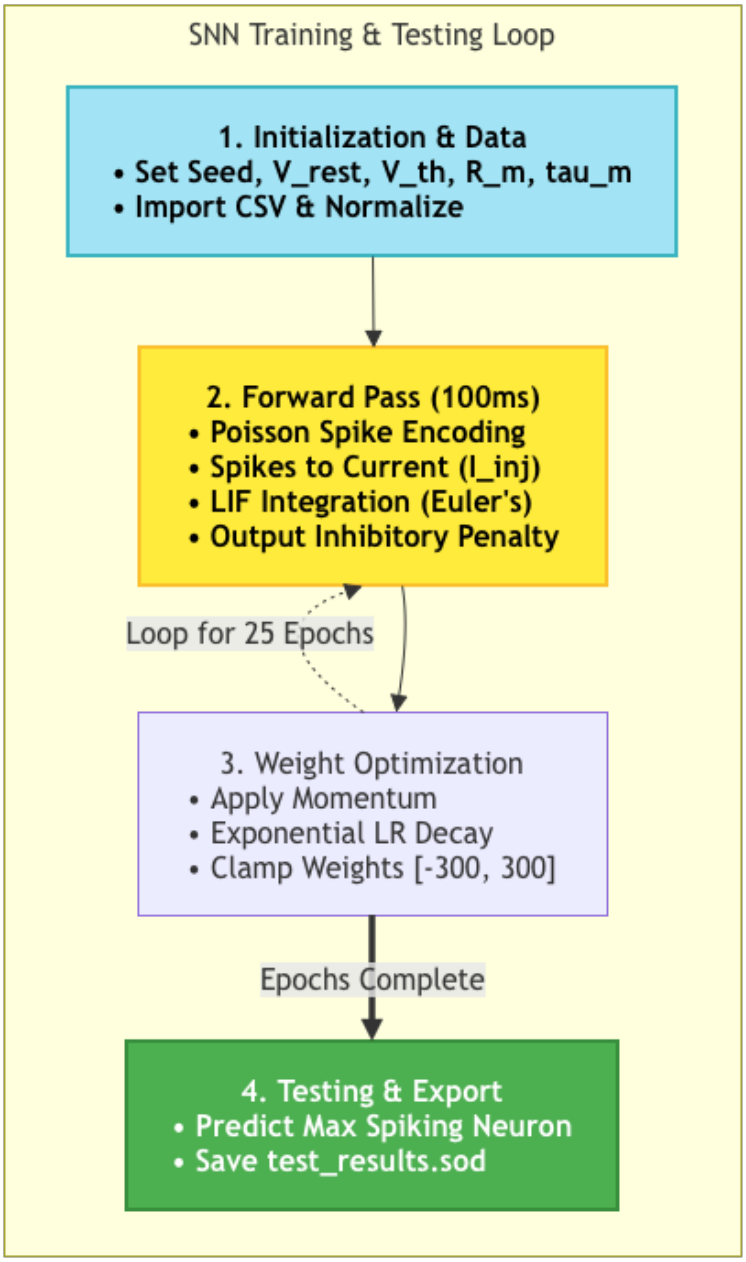
$$V[t+1] = V_{rest} + \beta_{decay}(V[t] - V_{rest}) + \frac{I_{inj}[t] \cdot R_m \cdot dt}{\tau_m} \tag{2.1}$$

- **Spike-Perceptron Weight Update:** Updating all 640 connections in a single matrix calculation using learning rate η and Error Transpose (E^T):

$$\Delta W = \eta(Inputs \times E^T) \tag{2.2}$$

- **Momentum Optimization:** Accumulated past weight updates prevent the network from stalling in local minima.

2.4 Flowchart



2.5 Results and Statistical Analysis

The model processed the test dataset achieving an accuracy of 91.90%. The proximity of the Macro F1-Score (92.03%) to the Global Accuracy proves the network learned highly balanced geometric representations across all ten digit classes without overfitting.

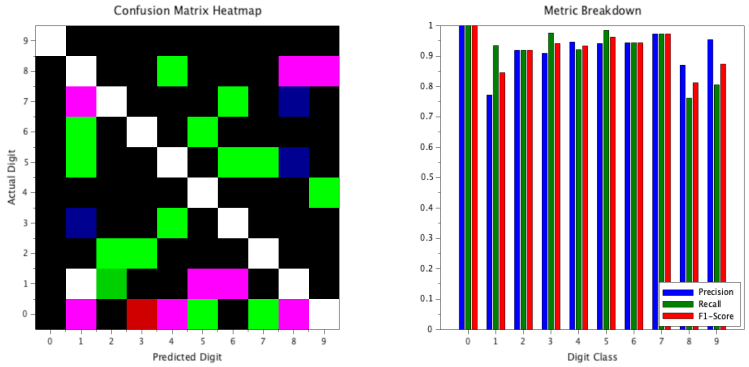


Figure 2.1: Confusion Matrix Heatmap showing accurate classification across digits 0-9.

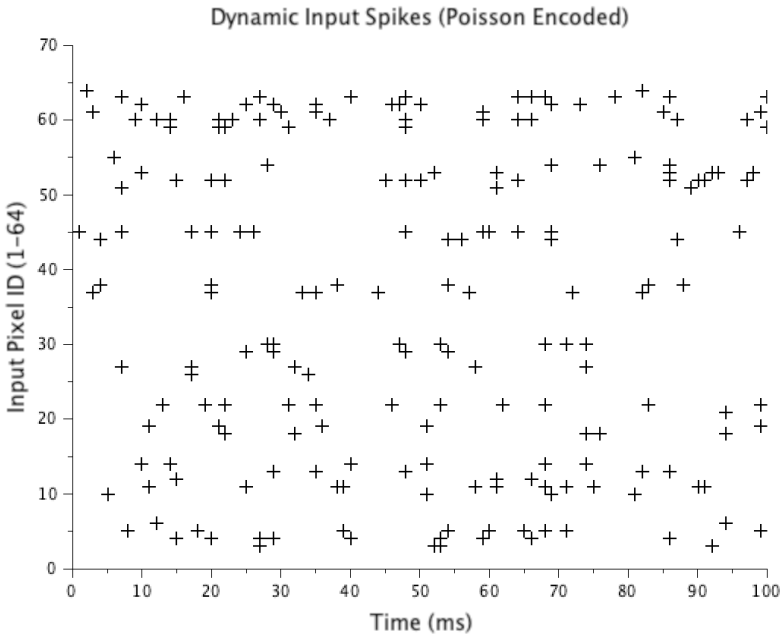


Figure 2.2: Raster plot of Poisson-encoded input spikes over the 100 ms timeframe.

Chapter 3

Case Study 2: Continuous-Time Simulation of Neural Synapses in Xcos

3.1 Abstract

This project simulates both inhibitory and excitatory chemical synapses using the fundamental potential and current equations of the Hodgkin-Huxley (HH) model. The simulation was built entirely within base Scilab/Xcos using a modular, continuous-time block architecture and superblocks. To accurately model synaptic transmission, first-order kinetic equations were implemented rather than artificial square-wave pulse generators. By adjusting the synaptic reversal potential, we successfully demonstrated both Excitatory Post-Synaptic Potentials (EPSP) that actively drive target neurons to spike and Inhibitory Post-Synaptic Potentials (IPSP) that suppress firing.

3.2 Problem Formulation

The standard Leaky Integrate-and-Fire (LIF) model simplifies the neuronal membrane into a basic RC circuit, lacking the dynamic memory elements found in real biology. To replicate biological functions, the Hodgkin-Huxley (HH) model utilizes voltage-gated channels to control the flow of current. When information is transferred from one neuron to another, it occurs through a complex chemical connection that either spikes the target neuron (EPSP) or suppresses it (IPSP). This case study models these complex connections by injecting a synaptic current (I_{syn}) governed by first-order kinetics inside Xcos.

3.3 Basic Concepts

- **Membrane Voltage (V_{mem}) Integrator:** Acts as the "memory" of the cell's electrical state. It continuously integrates the rate of change:

$$\frac{dV}{dt} = \frac{I_{in} - I_{Na} - I_K - I_L}{C_M} \tag{3.1}$$

- **Gating Kinetics (n, m, h):** The flow of electricity is governed by the opening and closing of ion channels, modeled by differential equations (e.g., Potassium Activation n):

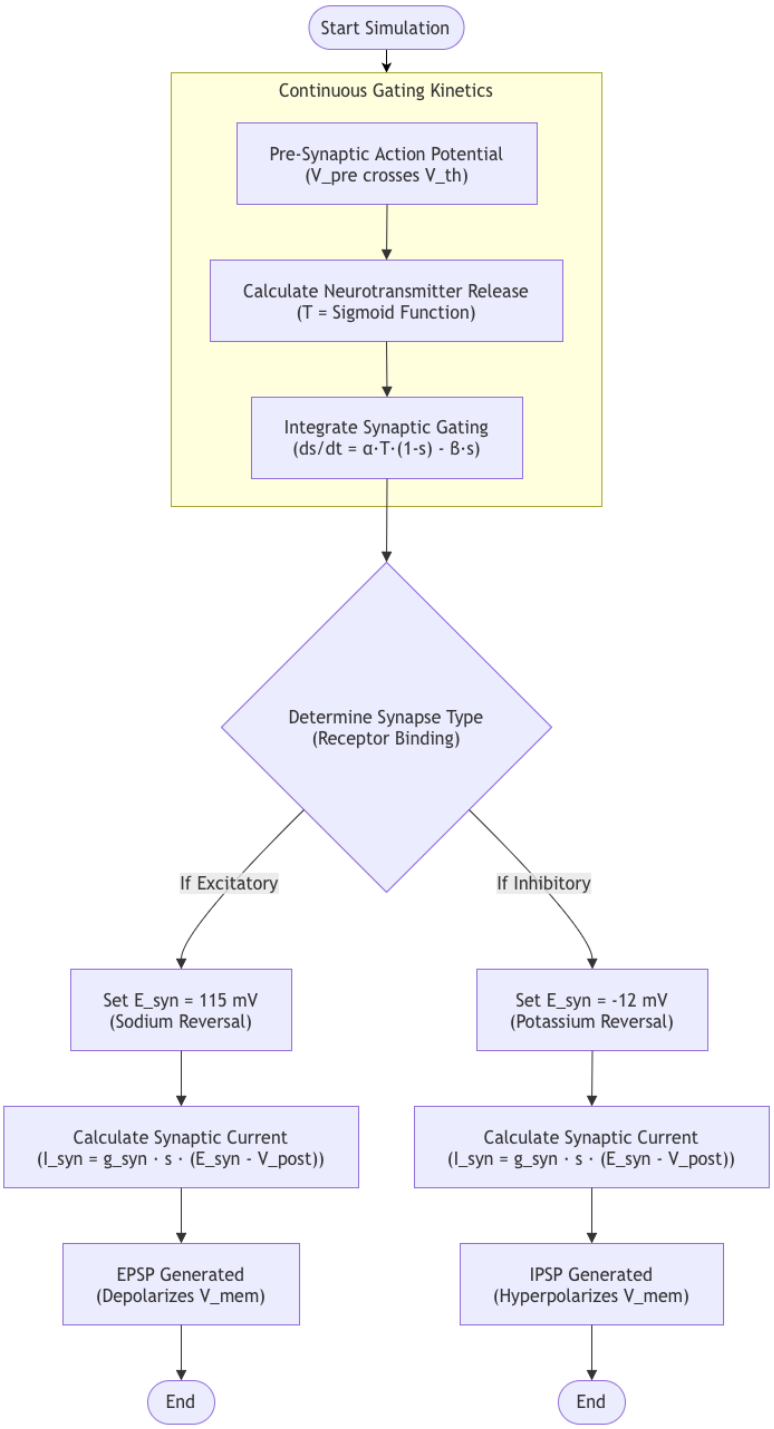
$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \tag{3.2}$$

- **Neurotransmitter Release (T):** Modeled using a continuous sigmoidal function to mimic a physical wave of neurotransmitters:

$$T = \frac{1}{1 + \exp(-(V_{pre} - V_{th})/K_p)} \tag{3.3}$$

- **Synaptic Current (I_{syn}):** The final current injected into the post-synaptic neuron depends on the reversal potential (E_{syn}). Setting $E_{syn} = 115$ mV creates an Excitatory synapse (EPSP), while $E_{syn} = -12$ mV creates an Inhibitory synapse (IPSP).

3.4 Flowchart



3.5 Xcos Simulation Architecture

The Neuron superblock relies on a continuous feedback loop between algebraic calculators (`sci_func`) and integrators. The Synapse superblock acts as a bridge connecting two independent HH neurons.

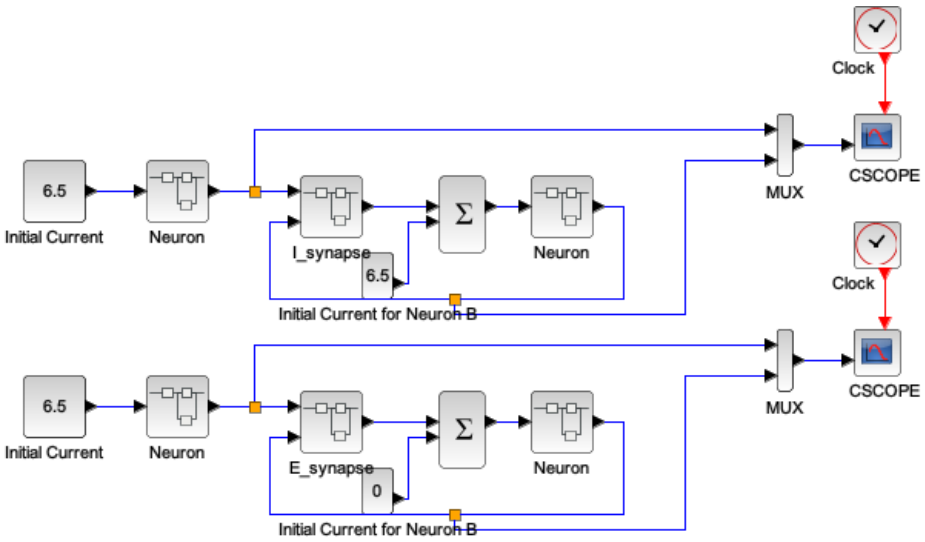


Figure 3.1: Xcos block diagram showing continuous-time integration of HH neurons and synaptic superblocks.

3.6 Results and Discussion

The simulation was executed over a 100 ms timeframe. In the EPSP configuration, the wave of neurotransmitters opened post-synaptic channels, depolarizing the target neuron and actively forcing it past the 20 mV threshold to trigger an action potential.

In the IPSP configuration, each time the pre-synaptic neuron fired, inhibitory channels opened, injecting a negative current that dragged the target neuron's membrane potential into a hyperpolarized state, continuously suppressing its firing.

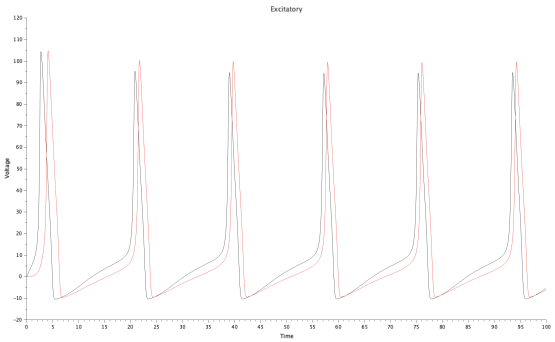


Figure 3.2: Excitatory Synapse (EPSP)

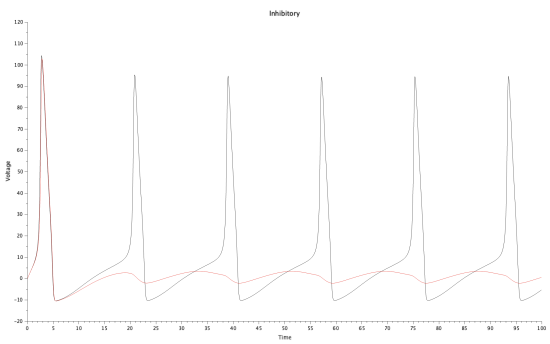


Figure 3.3: Inhibitory Synapse (IPSP)

Chapter 4

Case Study 3: Simulating the Izhikevich Model and Cortical Rhythms in Scilab

4.1 Abstract

This study recreates both single-cell membrane and large-scale cortical network behaviors based on Eugene M. Izhikevich's 2003 "Simple Model of Spiking Neurons." We implemented eight key cortical and thalamic firing patterns with a 0.1 ms Euler integration step. For real-time experimentation, we developed an interactive Graphical User Interface (GUI) dashboard offering live mathematical feedback. Additionally, a 1,000-neuron pulse-coupled neural network (PCNN) was built using a 1000×1000 synaptic weight matrix. Using a 0.5 ms split-step integration method and stochastic thalamic noise, the simulated network successfully self-organized to replicate mammalian alpha (10 Hz) and gamma (40 Hz) brain rhythms.

4.2 The Izhikevich Model

The model reduces Hodgkin-Huxley dynamics into a 2-D system of ordinary differential equations:

$$v' = 0.04v^2 + 5v + 140 - u + I \tag{4.1}$$

$$u' = a(bv - u) \tag{4.2}$$

with the after-spike resetting condition:

$$\text{if } v \geq 30 \text{ mV, then } v \leftarrow c,\ u \leftarrow u + d \tag{4.3}$$

4.3 GUI Dashboard Architecture

A robust, professional UI was constructed in Scilab. The dashboard prevents invalid handle crashes and handles real-time parameter tuning with dynamic equation substitutions directly rendered onto the axis titles.

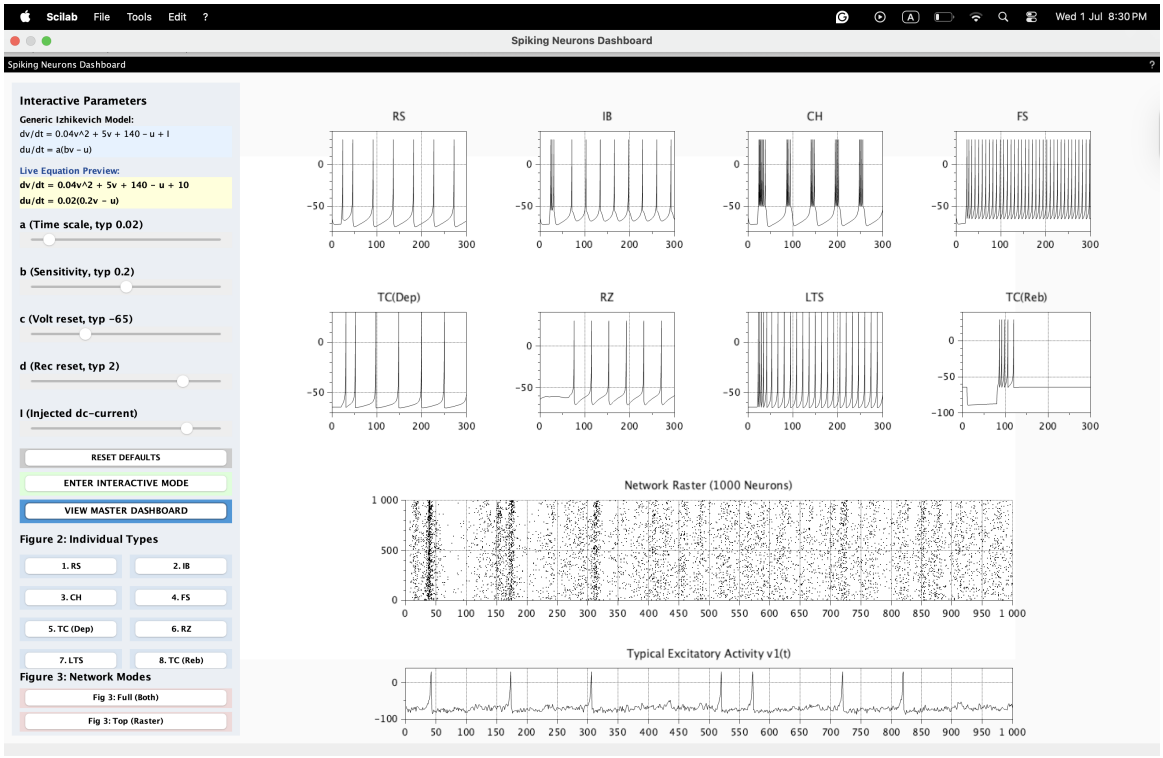
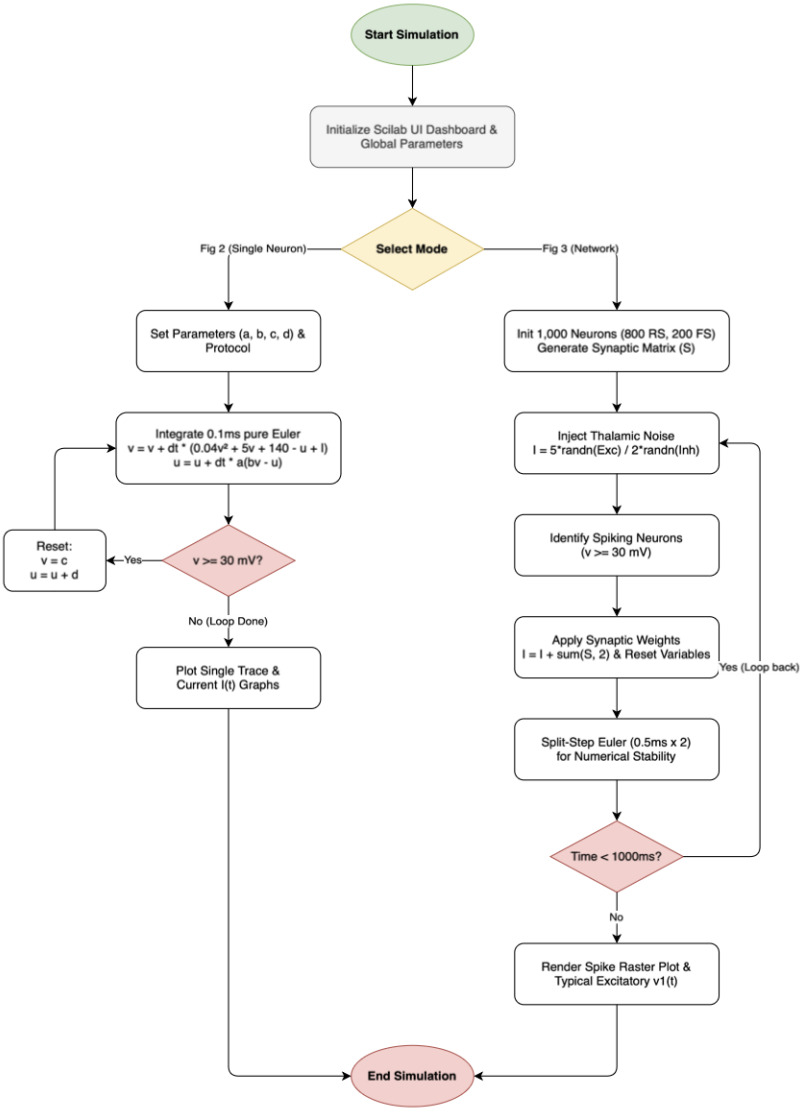


Figure 4.1: The Interactive Scilab Spiking Neurons Master Dashboard.

4.4 Flowchart



4.5 Results and Validation

4.5.1 Single Neuron Firing Modes

- **Regular Spiking (RS) & Chattering (CH):** Fine-tuning the recovery reset parameter ($d = 2.1$) for the Chattering profile reproduced its characteristic high-frequency multi-spike bursting pattern. The modified reset step acted as an internal recovery mechanism that sustained bursting under constant current injection without altering the baseline input.

- **Thalamo-Cortical (TC) Firing:** Initializing the membrane potential at the calculated resting equilibrium (-64.4 mV) reduced transient startup artifacts before stimulus delivery. Applying a hyperpolarizing current (-29.51) drove the neuron to approximately -87 mV, producing a clear post-inhibitory rebound burst upon current removal.

- **Resonator (RZ) Dynamics:** A holding current of 0.26 positioned the model just below the calculated Hopf bifurcation threshold (0.2625), allowing sustained subthreshold oscillations with minimal damping. A brief 2 ms, +1.0 trigger pulse pushed the system beyond the bifurcation threshold and initiated persistent repetitive spiking within the simulated time window, demonstrating the expected bistable behavior.

- **Intrinsically Bursting (IB):** Using a higher reset voltage ($c = -55$ mV) and moderate recovery reset ($d = 4$) reproduced an initial burst of closely spaced spikes followed by tonic spiking. Accumulation of the recovery variable (u) provided the negative feedback responsible for the transition from bursting to regular firing.

- **Fast Spiking (FS):** A rapid recovery rate ($a = 0.1$) enabled the recovery variable to closely track membrane dynamics. Under sustained depolarization, the neuron produced a high-frequency spike train with minimal spike-frequency adaptation, consistent with fast-spiking cortical interneurons.

- **Low-Threshold Spiking (LTS):** Increasing the sensitivity parameter ($b = 0.25$) lowered the effective firing threshold. Under depolarization, the neuron generated a rapid spike train exhibiting pronounced spike-frequency adaptation, with progressively increasing inter-spike intervals over time.

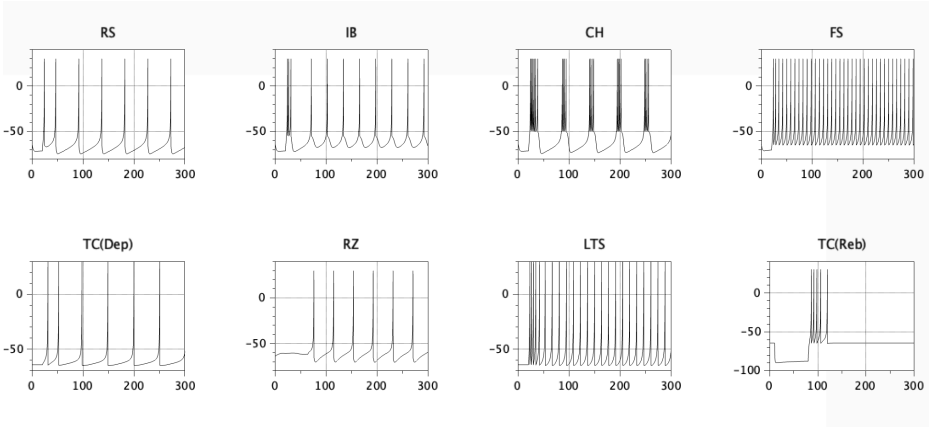


Figure 4.2: All 8 firing patterns

4.5.2 Pulse-Coupled Neural Network

We executed a 1,000-neuron model (800 excitatory, 200 inhibitory) utilizing a 0.5 ms dual split-step integration loop inside the 1.0 ms timeline for numerical stability. Driven by feedback loops and stochastic thalamic noise, the network successfully self-organized, producing dense vertical alignment bands in the raster plot corresponding to 10 Hz and 40 Hz synchronization.

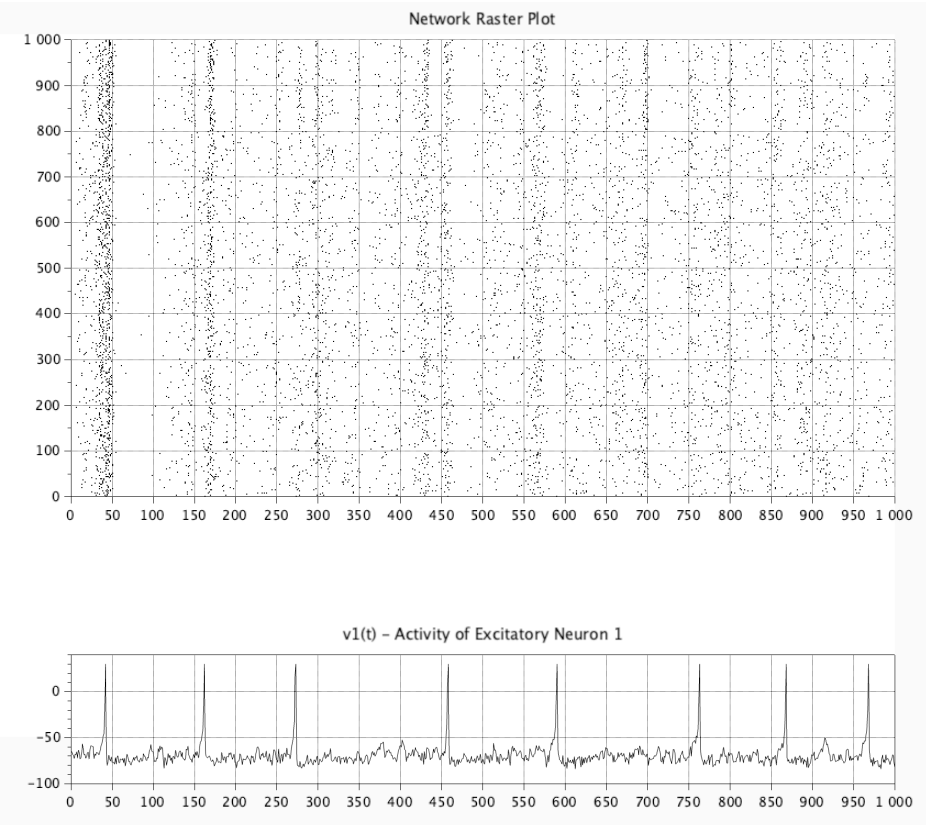


Figure 4.3: Network raster plot of 1,000 neurons self-organizing into rhythmic bands.

Chapter 5

Learnings

During this fellowship, I gained practical experience in using Scilab and Xcos for solving engineering problems through simulation, modelling, graphical analysis, and GUI development. The work helped me understand how theoretical concepts can be converted into computational case studies using open-source software.

From the first case study, I learned about Euler's integration and the basics of LIF equations. I learned about converting static pixels into firing probabilities and the target learning approach.

From the second case study, I learned more about biological concepts, such as neuron classes and gating variables. The HH model is more biologically accurate than LIF, but slower. I also learned about the superblock concept in Xcos.

From the third case study, Izhikevich's model introduces the interesting mathematical concept of bifurcations and treats neurons as a type of bifurcation. The GUI concepts inside Scilab were worth exploring, which could help in better simulation.

Overall, the fellowship improved my skills in Scilab programming, Xcos modeling, numerical simulation, GUI-based engineering computation, result interpretation, and technical documentation.

Chapter 6

Conclusion

This summer fellowship provided an enriching opportunity to explore and implement advanced neurocomputational concepts using base Scilab. Across the three progressive case studies, the versatility of Scilab as a free and open-source platform for modelling, simulation, continuous-time integration, and GUI development has been unequivocally demonstrated.

By beginning with custom supervised learning in Spiking Neural Networks, progressing to continuous-time biological first-order kinetics in Xcos, and finally building an interactive real-time dashboard for the canonical Izhikevich model, the theoretical mechanics of computational neuroscience were practically realized. These studies emphasize the impact of FOSS tools in bridging biological theory and mathematical practice, providing a cost-effective platform for advanced education and research without the need for proprietary software.

References

[1] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 7, pp. 1659-1671, 1997.

[2] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, art. no. 99, 2015.

[3] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge University Press, 2002.

[4] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting," *Neural Computation*, vol. 22, no. 2, pp. 467-510, 2010.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.

[6] G. G. Turrigiano, "The self-tuning neuron: synaptic scaling of excitatory synapses," *Cell*, vol. 135, no. 3, pp. 422-435, 2008.

[7] T. Yu, T. J. Sejnowski, and G. Cauwenberghs, "Biophysical neural spiking, bursting, and excitability dynamics in reconfigurable analog VLSI," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 5, pp. 420-429, Oct. 2011.

[8] A. Natarajan and J. Hasler, "Hodgkin-Huxley neuron and FPAA dynamics," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 4, pp. 918-926, Aug. 2018.

[9] A. Natarajan and J. Hasler, "Implementation of synapses with Hodgkin Huxley neurons on the FPAA," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo, Japan, May 2019, pp. 1-5.

[10] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, pp. 500-544, 1952.

[11] E. M. Izhikevich, "Simple Model of Spiking Neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569-1572, Nov. 2003.